I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 24 November 2022

S. Hares, Ed.
Huawei
J. Jeong, Ed.
J. Kim
Sungkyunkwan University
R. Moskowitz
HTT Consulting
Q. Lin
Huawei

23 May 2022

I2NSF Capability YANG Data Model
draft-ietf-i2nsf-capability-data-model-32

Abstract

This document defines an information model and the corresponding YANG data model for the capabilities of various Network Security Functions (NSFs) in the Interface to Network Security Functions (I2NSF) framework to centrally manage the capabilities of the various NSFs.

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 24 November 2022.

Table of Contents

1. Introduction

As the industry becomes more sophisticated and network devices (e.g., Internet-of-Things (IoT) devices, autonomous vehicles, and smartphones using Voice over Internet Protocol (VoIP) and Voice over Cellular Network, such as LTE and 5G (VoCN)) require advanced security protection in various scenarios, security service providers have a lot of problems described in [RFC8192] to provide such network devices with efficient and reliable security services in network

infrastructure. To resolve these problems, this document specifies the information and data models of the capabilities of Network Security Functions (NSFs) in a framework of the Interface to Network Security Functions (I2NSF) [RFC8329].

NSFs produced by multiple security vendors provide various security capabilities to customers. Multiple NSFs can be combined to provide security services over the given network traffic, regardless of whether the NSFs are implemented as physical or virtual functions. Security Capabilities describe the functions that Network Security Functions (NSFs) can provide for security policy enforcement. Security Capabilities are independent of the actual security policy that will implement the functionality of the NSF.

Every NSF should be described with the set of capabilities it offers. Security Capabilities enable security functionality to be described in a vendor-neutral manner. Security Capabilities are a market enabler, providing a way to define customized security protection by unambiguously describing the security features offered by a given NSF. Note that this YANG data model forms the basis of the NSF Monitoring Interface YANG data model [I-D.ietf-i2nsf-nsf-monitoring-data-model] and the NSF-Facing Interface YANG data model [I-D.ietf-i2nsf-nsf-facing-interface-dm].

This document provides an information model and the corresponding YANG data model [RFC6020][RFC7950] that defines the capabilities of NSFs to centrally manage the capabilities of those NSFs. The NSFs can register their own capabilities into a Network Operator Management (Mgmt) System (i.e., Security Controller) with this YANG data model through the registration interface [RFC8329]. With the database of the capabilities of those NSFs that are maintained centrally, those NSFs can be more easily managed [RFC8329].

This YANG data model uses an "Event-Condition-Action" (ECA) policy model that is used as the basis for the design of I2NSF Policy as described in [RFC8329] and Section 3.1. This policy model is not entirely perfect in which a conflict may happen between the configured policies, thus the YANG data model also provides an additional element of conflict resolution as described in Section 3.2. The "ietf-i2nsf-capability" YANG module defined in this document provides the following features:

* Definition for event capabilities of network security functions.

* Definition for condition capabilities of network security functions.

* Definition for action capabilities of network security functions.

* Definition for resolution strategy capabilities of network security functions.

* Definition for default action capabilities of network security functions.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terminology described in [RFC8329].

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

3. Requirements of I2NSF NSF Capability

This section provides the I2NSF Capability Information Model (CapIM). A CapIM is a formalization of the functionality that an NSF advertises. This enables the precise specification of what an NSF can do in terms of security policy enforcement, so that computer-based tasks can unambiguously refer to, use, configure, and manage NSFs. Capabilities are defined in a vendor- and technology-independent manner (i.e., regardless of the differences among vendors and individual products).

Network security experts can refer to categories of security controls and understand each other. For instance, network security experts agree on what is meant by the terms "NAT", "filtering", and "VPN concentrator". As a further example, network security experts unequivocally refer to "packet filters" as devices that allow or deny packet forwarding based on various conditions (e.g., source and destination IP addresses, source and destination ports, and IP protocol type fields) [Alshaer].

However, more information is required in case of other devices, like stateful firewalls or application layer filters. These devices filter packets or communications, but there are differences in the packets and communications that they can categorize and the states they maintain. Network engineers deal with these differences by asking more questions to determine the specific category and functionality of the device. Machines can follow a similar approach,

which is commonly referred to as question-answering [Hirschman]. In this context, the CapIM and the derived data model can provide important and rich information sources.

Analogous considerations can be applied for channel protection protocols, where we all understand that they will protect packets by means of symmetric algorithms whose keys could have been negotiated with asymmetric cryptography, but they may work at different layers and support different algorithms and protocols. To ensure protection, these protocols apply integrity, optionally confidentiality, anti-reply protections, and authentication.

The CapIM is intended to clarify these ambiguities by providing a formal description of NSF functionality. The set of functions that are advertised MAY be restricted according to the privileges of the user or application that is viewing those functions. I2NSF Capabilities enable unambiguous specification of the security capabilities available in a (virtualized) networking environment, and their automatic processing by means of computer-based techniques.

This CapIM enables a security controller in an I2NSF framework [RFC8329] to properly identify and manage NSFs, and allow NSFs to properly declare their functionality through a Developer's Management System (DMS) [RFC8329], so that they can be used in the correct way.

3.1. Design Principles and ECA Policy Model

This document defines an information model for representing NSF capabilities. Some basic design principles for security capabilities and the systems that manage them are:

* Independence: Each security capability (e.g., events, conditions, and actions) SHOULD be an independent function, with minimum overlap or dependency on other capabilities. This enables each security capability to be utilized and assembled with other security capabilities together freely. More importantly, changes to one capability SHOULD NOT affect other capabilities. This follows the Single Responsibility Principle [Martin] [OODSRP].

* Abstraction: Each capability MUST be defined in a vendor-independent manner.

* Advertisement: The Registration Interface [I-D.ietf-i2nsf-registration-interface-dm] MUST be used to advertise and register the capabilities of each NSF. This same interface MUST be used by other I2NSF Components to determine what Capabilities are currently available to them.

* Execution: The NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm] and NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] MUST be used to configure the use of a capability into an NSF and monitor the NSF, respectively. These provide a standardized ability to describe its functionality, and report its processing results, respectively. These facilitate multivendor interoperability.

* Automation: The system MUST have the ability to auto-discover, auto-negotiate, and auto-update the information of an NSF's registered security capabilities without human intervention. These features are especially useful for the management of a large number of NSFs. They are essential for adding smart services (e.g., refinement, analysis, capability reasoning, and optimization) to the security scheme employed. These features are supported by many design patterns, including the Observer Pattern [OODOP], the Mediator Pattern [OODMP], and a set of Message Exchange Patterns [Hohpe]. The Registration Interface [I-D.ietf-i2nsf-registration-interface-dm] can register the capabilities of NSFs with the security controller from the request of a Developer's Management System, providing a list of available NSFs, the corresponding security capabilities, and access information to the security controller. Also, this interface can send a query to Developer's Management System in order to find an NSF to satisfy the requested security capability from the security controller that receives a security policy.

* Scalability: The management system SHOULD have the capability to scale up/down or scale in/out. Thus, it can meet various performance requirements derived from changeable network traffic or service requests. In addition, security capabilities that are affected by scalability changes SHOULD support reporting statistics to the security controller to assist its decision on whether it needs to invoke scaling or not. The NSF Monitoring Interface [I-D.ietf-i2nsf-nsf-monitoring-data-model] can observe the performance of NSFs to let the security controller decide scalability changes of the NSFs.

Based on the above principles, this document defines a capability model that enables an NSF to register (and hence advertise) its set of capabilities that other I2NSF Components can use. These capabilities MUST have their access control restricted by a policy and the mechanism of access control is RECOMMENDED to follow the mechanism described in Network Configuration Access Control Model (NACM) [RFC8341]; the policy that determines which components are granted which access is out of scope for this document. The set of capabilities provided by a given set of NSFs defines the security services offered by the set of NSFs used. The security controller

can compare the requirements of users and applications with the set of capabilities that are currently available in order to choose which capabilities of which NSFs are needed to meet those requirements. Note that this choice is independent of vendor, and instead relies specifically on the capabilities (i.e., the description) of the functions provided.

Furthermore, NSFs are subject to the updates of security capabilities and software to cope with newly found security attacks or threats, hence new capabilities may be created, and/or existing capabilities may be updated (e.g., by updating its signature and algorithm). New capabilities may be sent to and stored in a centralized repository, or stored separately in a vendor's local repository. In either case, the Registration Interface can facilitate this update process so the Developer's Management System can let the security controller update its repository for NSFs and their security capabilities.

The "Event-Condition-Action" (ECA) policy model in [RFC8329] is used as the basis for the design of the capability model; The following three terms define the structure and behavior of an I2NSF imperative policy rule:

* Event: 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 I2NSF Policy Rules, it is used to determine whether the condition clause of an I2NSF 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 an ACL).

* Condition: 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 with a desired state.

* Action: An action is used to control and monitor aspects of NSFs to handle packets or flows 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 filtering (i.e., URL filtering) and flow filtering, and deep packet inspection for packets and flows.

An I2NSF Policy Rule is made up of three clauses: an Event clause, a Condition clause, and an Action clause. This structure is also called an ECA (Event-Condition-Action) Policy Rule. A Boolean clause is a logical statement that evaluates to either TRUE or FALSE. It may be made up of one or more terms; if more than one term is present, then each term in the Boolean clause is combined using logical connectives (i.e., AND, OR, and NOT).

An I2NSF ECA Policy Rule has the following semantics:

```
IF <event-clause> is TRUE

   IF <condition-clause> is TRUE

      THEN execute <action-clause> [constrained by metadata]

   END-IF

END-IF
```

Technically, the "Policy Rule" is really a container that aggregates the above three clauses, as well as metadata which describe the characteristics and behaviors of a capability (or an NSF). One example of metadata that has been well-associated with a network access control list is priority. Priority information is usually given to a rule as a numerical value to control the execution order of the rules. Associating a priority value an ECA policy enables a business logic to be used to prescribe a behavior. For example, suppose that a particular ECA Policy Rule contains three actions (A1, A2, and A3 in order). Action A2 has a priority of 10; actions A1 and A3 have no priority specified. Then, metadata may be used to restrict the set of actions that can be executed when the event and condition clauses of this ECA Policy Rule are evaluated to be TRUE; two examples are: (1) only the first action (A1) is executed, and then the policy rule returns to its caller, or (2) all actions are executed, starting with the highest priority.

The above ECA policy model is very general and easily extensible.

For example, when an NSF has both url filtering capability and packet filtering capability for protocol headers, it means that it can match the URL as well as the Ethernet header, IP header, and Transport header for packet filtering. The condition capability for url filtering and packet filtering is not tightly linked to the action capability due to the independence of our ECA design principle. The action capability only lists the type of action that the NSF can take to handle the matched packets.

3.2. Conflict, Resolution Strategy and Default Action

Formally, two I2NSF Policy Rules conflict with each other if:

* the Event Clauses of each evaluate to TRUE;

* the Condition Clauses of each evaluate to TRUE;

* the Action Clauses affect the same object in different ways.

For example, if we have two Policy Rules called R1 and R2 in the same Policy:

R1: During 8am-6pm, if traffic is external, then run through firewall

R2: During 7am-8pm, run antivirus

There is no conflict between the two policy rules R1 and R2, since the policy rules act on different conditions, where firewall verifies the packet header while antivirus verifies the contents. However, consider these two rules called R3 and R4:

R3: During 9am-6pm, allow John to access social networking service websites

R4: During 9am-6pm, disallow all users to access social networking service websites

The two policy rules R3 and R4 are now in conflict, between the hours of 9am and 6pm, because the actions of R3 and R4 are different and apply to the same user (i.e., John).

Conflicts theoretically compromise the correct functioning of devices. However, NSFs have been designed to cope with these issues. Since conflicts are originated by simultaneously matching rules, an additional process decides the action to be applied, e.g., among the actions which the matching rule would have enforced. This process is described by means of a resolution strategy for conflicts. The finding and handling of conflicted matching rules is performed by resolution strategies.

Some concrete examples of a resolution strategy are:

* First Matching Rule (FMR)

* Last Matching Rule (LMR)

* Prioritized Matching Rule (PMR) with Errors (PMRE)

* Prioritized Matching Rule with No Errors (PMRN)

In the above, a PMR strategy is defined as follows:

1. Order all actions by their Priority (highest is first, no priority is last); actions that have the same priority may be appear in any order in their relative location.

2. For PMRE: if any action fails to execute properly, temporarily stop the execution of all actions. Invoke the error handler of the failed action. If the error handler is able to recover from the error, then continue the execution of any remaining actions; else, terminate the execution of the ECA Policy Rule having those all actions.

3. For PMRN: if any action fails to execute properly, stop the execution of all actions. Invoke the error handler of the failed action, but regardless of the result, the execution of the ECA Policy Rule having those all actions MUST be terminated.

On the other hand, it may happen that, if an event is caught, none of the policy rules matches the condition. Note that a packet or flow is handled only when it matches both the event and condition of a policy rule according to the ECA policy model. As a simple case, no condition in the rules may match a packet arriving at the border firewall. In this case, the packet is usually dropped, that is, the firewall has a default behavior of packet dropping in order to manage the cases that are not covered by specific rules.

Therefore, this document introduces two further capabilities for an NSF to handle security policy conflicts with resolution strategies and enforce a default action if no rules match.

* Resolution Strategies: They can be used to specify how to resolve conflicts that occur between the actions of the similar or different policy rules that are matched and contained in this particular NSF; note that a badly written policy rule may cause a conflict of actions with another similar policy rule.

* Default Action: It provides the default behavior to be executed when there are no other alternatives. This action can be either an explicit action or a set of actions.

4. Overview of YANG Data Model

This section provides an overview of how the YANG data model can be used in the I2NSF framework described in [RFC8329]. Figure 1 shows the capabilities (e.g., firewall and web filter) of NSFs in the I2NSF Framework. As shown in this figure, a Developer's Management System (DMS) can register NSFs and their capabilities with a Security Controller. To register NSFs in this way, the DMS utilizes the standardized capability YANG data model in this document through the I2NSF Registration Interface [RFC8329]. That is, this Registration Interface uses the YANG module described in this document to describe the capabilities of an NSF that is registered with the Security Controller. As described in [RFC8192], with the usage of the Registration Interface and the YANG module in this document, the capabilities registration of NSFs manufactured by multiple vendors can be done together by the Security Controller in a centralized way, and the information of the registered Capabilities in the Security Controller information should be updated dynamically by each vendor as the NSF may have software or hardware updates.

In Figure 1, a new NSF at a Developer's Management System has capabilities of Firewall (FW) and Web Filter (WF), which are denoted as (Cap = {FW, WF}), to support Event-Condition-Action (ECA) policy rules where 'E', 'C', and 'A' mean "Event", "Condition", and "Action", respectively. The condition involves IPv4 or IPv6 datagrams, and the action includes "Allow" and "Deny" for those datagrams. Note that "E = {}" means that the event boolean will always evaluate to true.

Note that the NSF-Facing Interface [RFC8329] is used by the Security Controller to configure the security policy rules of NSFs (e.g., firewall and Distributed Denial-of-Service (DDoS) attack mitigator) with the capabilities of the NSFs registered with the Security Controller.

```
        +-------------------------------------------------------+
        |  I2NSF User (e.g., Overlay Network Mgmt, Enterprise   |
        |  Network Mgmt, another network domain's mgmt, etc.)   |
        +--------------------+----------------------------------+
            I2NSF            ^
    Consumer-Facing Interface|
                             |
                             v                 I2NSF
           +-----------------+------------+ Registration +-------------+
           | Network Operator Mgmt System |  Interface   | Developer's |
           | (i.e., Security Controller)  |<------------>| Mgmt System |
           +-----------------+------------+              +-------------+
                             ^                                New NSF
                             |                           Cap = {FW, WF}
               I2NSF         |                           E = {}
        NSF-Facing Interface |                           C = {IPv4, IPv6}
                             |                           A = {Allow, Deny}
                             v
        +---------------+----+------------+-----------------+
        |               |                 |                 |
    +---+---+       +---+---+         +---+---+         +---+---+
    | NSF-1 |  ...  | NSF-m |         | NSF-1 |   ...   | NSF-n |
    +-------+       +-------+         +-------+         +-------+
      NSF-1           NSF-m             NSF-1             NSF-n
    Cap = {FW, WF}   Cap = {FW, WF}   Cap = {FW, WF}    Cap = {FW, WF}
    E = {}           E = {user}       E = {dev}         E = {}
    C = {IPv4}       C = {IPv6}       C = {IPv4, IPv6}  C = {IPv4, time}
    A = {Allow,Deny} A = {Allow,Deny} A = {Allow,Deny}  A = {Allow,Deny}

     Developer's Mgmt System A          Developer's Mgmt System B
```

Figure 1: Capabilities of NSFs in I2NSF Framework

A use case of an NSF with the capabilities of firewall and web filter is described as follows.

* If a network administrator wants to apply security policy rules to block malicious users with firewall and web filter, it is a tremendous burden for a network administrator to apply all of the needed rules to NSFs one by one. This problem can be resolved by managing the capabilities of NSFs as described in this document.

* If a network administrator wants to block IPv4 or IPv6 packets from malicious users, the network administrator sends a security policy rule to the Network Operator Management System (i.e., Security Controller) using the I2NSF Consumer-Facing Interface, directing the system to block the users in question.

* When the Network Operator Management System receives the security policy rule, it automatically sends that security policy rule to appropriate NSFs (i.e., NSF-m in Developer's Management System A and NSF-1 in Developer's Management System B) which can support the capabilities (i.e., IPv6). This lets an I2NSF User not consider which specific NSF(s) will work for the security policy rule.

* If NSFs encounter the suspicious IPv4 or IPv6 packets of malicious users, they can filter the packets out according to the configured security policy rule. Therefore, the security policy rule against the malicious users' packets can be automatically applied to appropriate NSFs without human intervention.

5. YANG Tree Diagram

This section shows a YANG tree diagram of capabilities of network security functions, as defined in the Section 3.

5.1. Network Security Function (NSF) Capabilities

This section explains a YANG tree diagram of NSF capabilities and its features. Figure 2 shows a YANG tree diagram of NSF capabilities. The NSF capabilities in the tree include directional capabilities, event capabilities, condition capabilities, action capabilities, resolution strategy capabilities, and default action capabilities. Those capabilities can be tailored or extended according to a vendor's specific requirements. Refer to the NSF capabilities information model for detailed discussion in Section 3.

```
module: ietf-i2nsf-capability
  +--rw nsf* [nsf-name]
     +--rw nsf-name                            string
     +--rw directional-capabilities*           identityref
     +--rw event-capabilities
     |  +--rw system-event-capability*   identityref
     |  +--rw system-alarm-capability*   identityref
     +--rw condition-capabilities
     |  +--rw generic-nsf-capabilities
     |  |  +--rw ethernet-capability*   identityref
     |  |  +--rw ipv4-capability*       identityref
     |  |  +--rw ipv6-capability*       identityref
     |  |  +--rw icmpv4-capability*     identityref
     |  |  +--rw icmpv6-capability*     identityref
     |  |  +--rw tcp-capability*        identityref
     |  |  +--rw udp-capability*        identityref
     |  |  +--rw sctp-capability*       identityref
     |  |  +--rw dccp-capability*       identityref
     |  +--rw advanced-nsf-capabilities
     |  |  +--rw anti-ddos-capability*              identityref
     |  |  +--rw ips-capability*                    identityref
     |  |  +--rw anti-virus-capability*             identityref
     |  |  +--rw url-filtering-capability*          identityref
     |  |  +--rw voip-vocn-filtering-capability*    identityref
     |  +--rw context-capabilities
     |     +--rw time-capabilities*                 identityref
     |     +--rw application-filter-capabilities*   identityref
     |     +--rw device-type-capabilities*          identityref
     |     +--rw user-condition-capabilities*       identityref
     |     +--rw geographic-capabilities*           identityref
     +--rw action-capabilities
     |  +--rw ingress-action-capability*   identityref
     |  +--rw egress-action-capability*    identityref
     |  +--rw log-action-capability*       identityref
     +--rw resolution-strategy-capabilities*   identityref
     +--rw default-action-capabilities*        identityref
```

Figure 2: YANG Tree Diagram of Capabilities of Network Security Functions

The data model in this document provides identities for the capabilities of NSFs. Every identity in the data model represents the capability of an NSF. Each identity is explained in the description of the identity.

Event capabilities are used to specify the capabilities that describe an event that would trigger the evaluation of the condition clause of the I2NSF Policy Rule. The defined event capabilities are system event and system alarm.

Condition capabilities are used to specify capabilities of 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 a set of actions needs to be executed or not so that an imperative I2NSF policy rule can be executed. In this document, two kinds of condition capabilities are used to classify different capabilities of NSFs such as generic-nsf-capabilities and advanced-nsf-capabilities. First, the generic-nsf-capabilities define NSFs that operate on packet header for layer 2 (i.e., Ethernet capability), layer 3 (i.e., IPv4 capability, IPv6 capability, ICMPv4 capability, and ICMPv6 capability.), and layer 4 (i.e., TCP capability, UDP capability, SCTP capability, and DCCP capability). Second, the advanced-nsf-capabilities define NSFs that operate on features different from the generic-nsf-capabilities, e.g., the payload, cross flow state, application layer, traffic statistics, network behavior, etc. This document defines the advanced-nsf into two categories such as content-security-control and attack-mitigation-control.

* Content security control is an NSF that evaluates the payload of a packet, such as Intrusion Prevention System (IPS), URL-Filtering, Antivirus, and VoIP (Voice over Internet Protocol) / VoCN (Voice over Cellular Network) Filter.

* Attack mitigation control is an NSF that mitigates an attack such as anti-DDoS (DDoS-mitigator).

The advanced-nsf can be extended with other types of NSFs. This document only provides five advanced-nsf capabilities, i.e., IPS capability, URL-Filtering capability, Antivirus capability, VoIP/VoCN Filter capability, and Anti-DDoS capability. Note that VoIP and VoCN are merged into a single capability in this document because VoIP and VoCN use the Session Initiation Protocol (SIP) [RFC3261] for a call setup. See Section 3.1 for more information about the condition in the ECA policy model. Also note that QUIC protocol [RFC9000] is excluded in the data model as it is not considered in the initial I2NSF documents [RFC8329]. The QUIC traffic should not be treated as UDP traffic and will be considered in the future I2NSF documents.

The context capabilities provide extra information for the condition. The given context conditions are application filter, target, user condition, and geographic location. Time capabilities are used to specify the capabilities which describe when to execute the I2NSF

policy rule. The time capabilities are defined in terms of absolute time and periodic time, where the absolute time means the exact time to start or end, and the periodic time means repeated time like day, week, month, or year. The application filter capability is the capability for matching the packet based on the application protocol, such as HTTP, HTTPS, FTP, etc. The device type capability is the capability for matching the type of the destination devices, such as PC, IoT, Network Infrastructure devices, etc. The user condition is the capability for matching the users of the network by mapping each user ID to an IP address. Users can be combined into groups. The geographic location capability is the capability for matching the geographical location of a source or destination of a packet.

Note that due to the exclusion of QUIC protocol in the I2NSF documents, HTTP/3 is also excluded in the document and will be considered in the future I2NSF documents along with the QUIC protocol. HTTP/3 should not be interpreted as either HTTP/1.1 or HTTP/2.

Action capabilities are used to specify the capabilities that describe the control and monitoring aspects of flow-based NSFs when the event and condition clauses are satisfied. The action capabilities are defined as ingress-action capability, egress-action capability, and log-action capability. See Section 3.1 for more information about the action in the ECA policy model. Also, see Section 7.2 (NSF-Facing Flow Security Policy Structure) in [RFC8329] for more information about the ingress and egress actions. In addition, see Section 9.1 (Flow-Based NSF Capability Characterization) in [RFC8329] and Section 6.5 (NSF Logs) in [I-D.ietf-i2nsf-nsf-monitoring-data-model] for more information about logging at NSFs.

Resolution strategy capabilities are used to specify the capabilities that describe conflicts that occur between the actions of the similar or different policy rules that are matched and contained in this particular NSF; note that a badly written policy rule may cause a conflict of actions with another similar policy rule. The resolution strategy capabilities are defined as First Matching Rule (FMR), Last Matching Rule (LMR), Prioritized Matching Rule with Error (PMRE), and Prioritized Matching with No Errors (PMRN). See Section 3.2 for more information about the resolution strategy.

Default action capabilities are used to specify the capabilities that describe how to execute I2NSF policy rules when no rule matches a packet. The default action capabilities are defined as pass, drop, reject, rate-limit, and mirror. See Section 3.2 for more information about the default action.

6. YANG Data Model of I2NSF NSF Capability

This section introduces a YANG module for NSFs' capabilities, as defined in the Section 3.

It makes references to

* [RFC0768]
* [RFC0791]
* [RFC0792]
* [RFC0854]
* [RFC0959]
* [RFC1939]
* [RFC2474]
* [RFC2595]
* [RFC3022]
* [RFC3168]
* [RFC3261]
* [RFC4250]
* [RFC4340]
* [RFC4443]
* [RFC4766]
* [RFC5103]
* [RFC5321]
* [RFC5595]
* [RFC6335]
* [RFC6437]
* [RFC6691]

* [RFC6864]

* [RFC7323]

* [RFC8075]

* [RFC8200]

* [RFC8311]

* [RFC8329]

* [RFC8805]

* [RFC9051]

* [IEEE802.3-2018]

* [IANA-Protocol-Numbers]

* [I-D.ietf-httpbis-http2bis]

* [I-D.ietf-httpbis-messaging]

* [I-D.ietf-httpbis-semantics]

* [I-D.ietf-tcpm-rfc793bis]

* [I-D.ietf-tcpm-accurate-ecn]

* [I-D.ietf-tsvwg-rfc4960-bis]

* [I-D.ietf-tsvwg-udp-options]

* [I-D.ietf-i2nsf-nsf-monitoring-data-model]

```
<CODE BEGINS> file "ietf-i2nsf-capability@2022-05-23.yang"
module ietf-i2nsf-capability {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability";
  prefix
    i2nsfcap;

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

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

        Editor: Susan Hares
        <mailto:shares@ndzh.com>

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

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

        Editor: Robert Moskowitz
        <mailto:rgm@htt-consult.com>

        Editor: Qiushi Lin
        <mailto:linqiushi@huawei.com>

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

     description
       "This module is a YANG module for I2NSF Network Security
        Functions (NSFs)'s Capabilities.

        Copyright (c) 2022 IETF Trust and the persons identified as
        authors of the code. All rights reserved.

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

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

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

     revision "2022-05-23"{
       description "Initial revision.";
       reference
         "RFC XXXX: I2NSF Capability YANG Data Model";
```

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

    /*
     * Identities
     */

    identity event {
      description
        "Base identity for I2NSF events.";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
         Monitoring Interface YANG Data Model - Event";
    }

    identity system-event {
      base event;
      description
        "Base identity for system event. System event (also called
         alert) is defined as a warning about any changes of
         configuration, any access violation, the information of
         sessions and traffic flows.";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
         Monitoring Interface YANG Data Model - System event";
    }

    identity system-alarm {
      base event;
      description
        "Base identity for system alarm. System alarm is defined as a
         warning related to service degradation in system hardware.";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
         Monitoring Interface YANG Data Model - System alarm";
    }

    identity access-violation {
      base system-event;
      description
        "Identity for access violation event. Access-violation system
         event is an event when a user tries to access (read, write,
         create, or delete) any information or execute commands
         above their privilege (i.e., not-conformant with the
         access profile).";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
```

```
          Monitoring Interface YANG Data Model - System event for access
          violation";
     }

     identity configuration-change {
       base system-event;
       description
         "Identity for configuration change event. Configuration change
          is a system event when a new configuration is added or an
          existing configuration is modified.";
       reference
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System event for
          configuration change";
     }

     identity memory-alarm {
       base system-alarm;
       description
         "Memory is the hardware to store information temporarily or for
          a short period, i.e., Random Access Memory (RAM). A
          memory-alarm is emitted when the memory usage is exceeding
          the threshold.";
       reference
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System alarm for
          memory";
     }

     identity cpu-alarm {
       base system-alarm;
       description
         "CPU is the Central Processing Unit that executes basic
          operations of the system. A cpu-alarm is emitted when the CPU
          usage is exceeding a threshold.";
       reference
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System alarm for CPU";
     }

     identity disk-alarm {
       base system-alarm;
       description
         "Disk or storage is the hardware to store information for a
          long period, i.e., Hard Disk and Solid-State Drive. A
          disk-alarm is emitted when the disk usage is exceeding a
          threshold.";
       reference
```

```
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System alarm for disk";
     }

     identity hardware-alarm {
       base system-alarm;
       description
         "A hardware alarm is emitted when a hardware failure (e.g.,
          CPU, memory, disk, or interface) is detected. A hardware
          failure is a malfunction within the electronic circuits or
          electromechanical components of the hardware that makes it
          unusable.";
       reference
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System alarm for
          hardware";
     }

     identity interface-alarm {
       base system-alarm;
       description
         "Interface is the network interface for connecting a device
          with the network. The interface-alarm is emitted when the
          state of the interface is changed.";
       reference
         "draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF NSF
          Monitoring Interface YANG Data Model - System alarm for
          interface";
     }

     identity time {
       description
         "Base identity for time capabilities";
     }

     identity absolute-time {
       base time;
       description
         "absolute time capabilities.
          If a network security function has the absolute time
          capability, the network security function supports
          rule execution according to absolute time.";
     }

     identity periodic-time {
       base time;
       description
         "periodic time capabilities.
```

```
          If a network security function has the periodic time
          capability, the network security function supports
          rule execution according to periodic time.";
     }

     identity device-type {
       description
         "Base identity for device type condition capability. The
          capability for matching the source or destination device
          type.";
     }

     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";
     }

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

     identity iot {
```

```
      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 user-condition {
      description
        "Base identity for user condition capability. This is the
         capability of mapping user(s) into their corresponding IP
         address";
    }

    identity user {
      base user-condition;
      description
        "Identity for user condition capability.
         A user (e.g., employee) can be mapped to an IP address of
         a computing device (e.g., computer, laptop, and virtual
         machine) which the user is using.";
    }

    identity group {
      base user-condition;
      description
        "Identity for group condition capability.
         A group (e.g., employees) can be mapped to multiple IP
```

```
          addresses of computing devices (e.g., computers, laptops,
          and virtual machines) which the group is using.";
     }

     identity geographic-location {
       description
         "Base identity for geographic location condition capability";
       reference
         "RFC 8805: A Format for Self-Published IP Geolocation Feeds -
          An access control for a geographical location (i.e.,
          geolocation) that has the corresponding IP prefix.";
     }

     identity source-location {
       base geographic-location;
       description
         "Identity for source geographic location condition capability";
       reference
         "RFC 8805: A Format for Self-Published IP Geolocation Feeds -
          An access control for a geographical location (i.e.,
          geolocation) that has the corresponding IP prefix.";
     }

     identity destination-location {
       base geographic-location;
       description
         "Identity for destination geographic location condition
          capability";
       reference
         "RFC 8805: A Format for Self-Published IP Geolocation Feeds -
          An access control for a geographical location (i.e.,
          geolocation) that has the corresponding IP prefix.";
     }

     identity directional {
       description
         "Base identity for directional traffic flow export capability";
       reference
         "RFC 5103: Bidirectional Flow Export Using IP Flow Information
          Export (IPFIX) - Terminology Unidirectional and Bidirectional
          Flow";
     }

     identity unidirectional {
       base directional;
       description
         "Identity for unidirectional traffic flow export.";
       reference
```

```
        "RFC 5103: Bidirectional Flow Export Using IP Flow Information
         Export (IPFIX) - Terminology Unidirectional Flow";
    }

    identity bidirectional {
      base directional;
      description
        "Identity for bidirectional traffic flow export.";
      reference
        "RFC 5103: Bidirectional Flow Export Using IP Flow Information
         Export (IPFIX) - Terminology Bidirectional Flow";
    }

    identity protocol {
      description
        "Base identity for protocols";
    }

    identity ethernet {
      base protocol;
      description
        "Base identity for Ethernet protocol.";
    }

    identity source-mac-address {
      base ethernet;
      description
        "Identity for the capability of matching Media Access Control
         (MAC) source address(es) condition capability.";
      reference
        "IEEE 802.3 - 2018: IEEE Standard for Ethernet";
    }

    identity destination-mac-address {
      base ethernet;
      description
        "Identity for the capability of matching Media Access Control
         (MAC) destination address(es) condition capability.";
      reference
        "IEEE 802.3 - 2018: IEEE Standard for Ethernet";
    }

    identity ether-type {
      base ethernet;
      description
        "Identity for the capability of matching the EtherType in
         Ethernet II and Length in Ethernet 802.3 of a packet.";
      reference
```

```
        "IEEE 802.3 - 2018: IEEE Standard for Ethernet";
    }

    identity ip {
      base protocol;
      description
        "Base identity for internet/network layer protocol,
         e.g., IPv4, IPv6, and ICMP.";
    }

    identity ipv4 {
      base ip;
      description
        "Base identity for IPv4 condition capability";
      reference
        "RFC 791: Internet Protocol";
    }

    identity ipv6 {
      base ip;
      description
        "Base identity for IPv6 condition capabilities";
      reference
        "RFC 8200: Internet Protocol, Version 6 (IPv6)
         Specification";
    }

    identity dscp {
      base ipv4;
      base ipv6;
      description
        "Identity for the capability of matching IPv4 annd IPv6
         Differentiated Services Codepoint (DSCP) condition";
      reference
        "RFC 791: Internet Protocol - Type of Service
         RFC 2474: Definition of the Differentiated
         Services Field (DS Field) in the IPv4 and
         IPv6 Headers
         RFC 8200: Internet Protocol, Version 6 (IPv6)
         Specification - Traffic Class";
    }

    identity ecn {
      base ipv4;
      base ipv6;
      description
        "Identity for the capability of matching IPv4 annd IPv6
         Explicit Congestion Notification (ECN) condition";
```

```
      reference
        "RFC 3168: The Addition of Explicit Congestion
         Notification (ECN) to IP.
         RFC 8311: Relaxing Restrictions on Explicit Congestion
         Notification (ECN) Experimentation";
    }

    identity total-length {
      base ipv4;
      base ipv6;
      description
        "Identity for the capability of matching IPv4 Total Length
         header field or IPv6 Payload Length header field.

         IPv4 Total Length is the length of datagram, measured in
         octets, including internet header and data.

         IPv6 Payload Length is the length of the IPv6 payload, i.e.,
         the rest of the packet following the IPv6 header, measured in
         octets.";
      reference
        "RFC 791: Internet Protocol - Total Length
         RFC 8200: Internet Protocol, Version 6 (IPv6)
         Specification - Payload Length";
    }

    identity ttl {
      base ipv4;
      base ipv6;
      description
        "Identity for the capability of matching IPv4 Time-To-Live
         (TTL) or IPv6 Hop Limit.";
      reference
        "RFC 791: Internet Protocol - Time To Live (TTL)
         RFC 8200: Internet Protocol, Version 6 (IPv6)
         Specification - Hop Limit";
    }

    identity next-header {
      base ipv4;
      base ipv6;
      description
        "Identity for the capability of matching IPv4 Protocol field
         and IPv6 Next Header field. Note that IPv4 Protocol field is
         equivalent to IPv6 Next Header field.";
      reference
        "IANA Website: Assigned Internet Protocol Numbers
         - Protocol Numbers
```

```
          RFC 791: Internet Protocol - Protocol
          RFC 8200: Internet Protocol, Version 6 (IPv6)
          Specification - Next Header";
     }

     identity source-address {
       base ipv4;
       base ipv6;
       description
         "Identity for the capability of matching IPv4 or IPv6 source
          address(es) condition capability.";
       reference
         "RFC 791: Internet Protocol - Address
          RFC 8200: Internet Protocol, Version 6 (IPv6)
          Specification - Source Address";
     }

     identity destination-address {
       base ipv4;
       base ipv6;
       description
         "Identity for the capability of matching IPv4 or IPv6
          destination address(es) condition capability.";
       reference
         "RFC 791: Internet Protocol - Address
          RFC 8200: Internet Protocol, Version 6 (IPv6)
          Specification - Destination Address";
     }

     identity flow-direction {
       base ipv4;
       base ipv6;
       description
         "Identity for flow direction of matching IPv4/IPv6 source
          or destination address(es) condition capability where a flow's
          direction is either unidirectional or bidirectional";
       reference
         "RFC 791: Internet Protocol
          RFC 8200: Internet Protocol, Version 6 (IPv6)
          Specification";
     }

     identity ihl {
       base ipv4;
       description
         "Identity for matching IPv4 header-length (IHL)
         condition capability";
       reference
```

```
        "RFC 791: Internet Protocol - Header Length";
    }

    identity identification {
      base ipv4;
      description
        "Identity for IPv4 identification condition capability.
         IPv4 ID field is used for fragmentation and reassembly.";
      reference
        "RFC 791: Internet Protocol - Identification
         RFC 6864: Updated Specification of the IPv4 ID Field -
         Fragmentation and Reassembly";
    }

    identity fragment-offset {
      base ipv4;
      description
        "Identity for matching IPv4 fragment offset
         condition capability";
      reference
        "RFC 791: Internet Protocol - Fragmentation Offset";
    }

    identity flow-label {
      base ipv6;
      description
        "Identity for matching IPv6 flow label
         condition capability";
      reference
        "RFC 8200: Internet Protocol, Version 6 (IPv6)
         Specification - Flow Label
         RFC 6437: IPv6 Flow Label Specification";
    }

    identity transport-protocol {
      base protocol;
      description
        "Base identity for Layer 4 protocol condition capabilities,
         e.g., TCP, UDP, SCTP, and DCCP";
    }

    identity tcp {
      base transport-protocol;
      description
        "Base identity for TCP condition capabilities";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification";
```

```
     }

     identity udp {
       base transport-protocol;
       description
         "Base identity for UDP condition capabilities";
       reference
         "RFC 768: User Datagram Protocol";
     }

     identity sctp {
       base transport-protocol;
       description
         "Base identity for SCTP condition capabilities";
       reference
         "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
          Protocol";
     }

     identity dccp {
       base transport-protocol;
       description
         "Base identity for DCCP condition capabilities";
       reference
         "RFC 4340: Datagram Congestion Control Protocol";
     }

     identity source-port-number {
       base tcp;
       base udp;
       base sctp;
       base dccp;
       description
         "Identity for matching TCP, UDP, SCTP, and DCCP source port
          number condition capability";
       reference
         "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
          (TCP) Specification
          RFC 768: User Datagram Protocol
          draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
          Protocol
          RFC 4340: Datagram Congestion Control Protocol";
     }

     identity destination-port-number {
       base tcp;
       base udp;
       base sctp;
```

```
      base dccp;
      description
        "Identity for matching TCP, UDP, SCTP, and DCCP destination
         port number condition capability";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification";
    }

    identity flags {
      base ipv4;
      base tcp;
      description
        "Identity for IPv4 flags and TCP control bits (flags) condition
         capability. Note that this should not be interpreted such that
         IPv4 flags and TCP flags are similar.
         If this identity is used under 'ipv4-capability', it indicates
         the support of matching the IPv4 flags header.
         If this identity is used under 'tcp-capability', it indicates
         the support of matching the TCP control bits (flags) header.
         The IPv4 flags is the three-bit field in IPv4 header to
         control and identify fragments.
         The TCP flags is the multiple one-bit fields after the
         reserved field in TCP header that indicates the connection
         states or provides additional information.";
      reference
        "RFC 791: Internet Protocol - Flags
         draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification - TCP Header Flags
         RFC 3168: The Addition of Explicit Congestion Notification
         (ECN) to IP - ECN-Echo (ECE) Flag and Congestion Window
         Reduced (CWR) Flag
         draft-ietf-tcpm-accurate-ecn-15: More Accurate ECN Feedback
         in TCP - ECN-Echo (ECE) Flag and Congestion Window Reduced
         (CWR) Flag";
    }

    identity options {
      base tcp;
      description
        "Identity for matching TCP options header field condition
         capability. When an NSF claims to have this capability, the
         NSF should be able to match the TCP options header field in
         binary.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification
         RFC 6691: TCP Options and Maximum Segment Size
```

```
          RFC 7323: TCP Extensions for High Performance";
     }

     identity data-offset {
       base tcp;
       base dccp;
       description
         "Identity for matching TCP and DCCP Data Offset condition
          capability.
          If this identity is used under 'tcp-capability', it indicates
          the support of matching the TCP data offset header.
          If this identity is used under 'sctp-capability', it indicates
          the support of matching the DCCP data offset header.
          The TCP Data Offset header field represents the size of the
          TCP header, expressed in 32-bit words.
          The DCCP Data Offset is the offset from the start of the
          packet's DCCP header to the start of its application data
          area, in 32-bit words.";
       reference
         "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
          (TCP) Specification - Data Offset
          RFC 4340: Datagram Congestion Control Protocol";
     }

     identity reserved {
       base tcp;
       description
         "Identity for TCP header reserved field condition capability.
          The set of control bits reserved for future used. The control
          bits are also known as flags. Must be zero in generated
          segments and must be ignored in received segments, if
          corresponding future features are unimplemented by the
          sending or receiving host.";
       reference
         "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
          (TCP) Specification";
     }

     identity window-size {
       base tcp;
       description
         "Identity for TCP header Window field condition capability.
          The number of data octets beginning with the one indicated
          in the acknowledgment field that the sender of this segment
          is willing to accept.";
       reference
         "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
          (TCP) Specification";
```

```
     }

     identity urgent-pointer {
       base tcp;
       description
         "Identity for TCP Urgent Pointer header field condition
          capability. The Urgent Pointer field in TCP describes the
          current value of urgent pointer as a positive offset from
          the sequence number in this segment. The urgent pointer
          points to the sequence number of the octet following the
          urgent data. This field is only be interpreted in segments
          with the URG control bit set.";
       reference
         "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
          (TCP) Specification";
     }

     identity length {
       base udp;
       base sctp;
       description
         "Identity for matching UDP length and SCTP chunk length
          condition capability.
          If this identity is used under 'udp-capability', it indicates
          the support of matching the UDP length header.
          If this identity is used under 'sctp-capability', it indicates
          the support of matching the SCTP chunk length header.
          The UDP length is the length in octets of this user datagram
          including this header and the datagram. The UDP length can be
          smaller than the IP transport length for UDP transport layer
          options.
          The SCTP chunk length represents the size of the chunk in
          bytes including the SCTP Chunk type, Chunk flags, Chunk flags,
          and Chunk Value fields.";
       reference
         "RFC 768: User Datagram Protocol - Length
          draft-ietf-tsvwg-udp-options: Transport Options for UDP
          draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
          Protocol - Chunk Length";
     }

     identity chunk-type {
       base sctp;
       description
         "Identity for SCTP chunk type condition capability";
       reference
         "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
          Protocol - Chunk Type";
```

```
     }

     identity service-code {
       base dccp;
       description
         "Identity for DCCP Service Code condition capability";
       reference
         "RFC 4340: Datagram Congestion Control Protocol
          RFC 5595: The Datagram Congestion Control Protocol (DCCP)
          Service Codes
          RFC 6335: Internet Assigned Numbers Authority (IANA)
          Procedures for the Management of the Service Name and
          Transport Protocol Port Number Registry - Service Code";
     }

     identity icmp {
       base protocol;
       description
         "Base identity for ICMPv4 and ICMPv6 condition capability";
       reference
         "RFC 792: Internet Control Message Protocol
          RFC 4443: Internet Control Message Protocol (ICMPv6)
          for the Internet Protocol Version 6 (IPv6) Specification
          - ICMPv6";
     }

     identity icmpv4 {
       base icmp;
       description
         "Base identity for ICMPv4 condition capability";
       reference
         "RFC 792: Internet Control Message Protocol";
     }

     identity icmpv6 {
       base icmp;
       description
         "Base identity for ICMPv6 condition capability";
       reference
         "RFC 4443: Internet Control Message Protocol (ICMPv6)
          for the Internet Protocol Ver sion 6 (IPv6) Specification
          - ICMPv6";
     }

     identity type {
       base icmpv4;
       base icmpv6;
       base dccp;
```

```
      description
        "Identity for ICMPv4, ICMPv6, and DCCP type condition
         capability";
      reference
        "RFC 792: Internet Control Message Protocol
         RFC 4443: Internet Control Message Protocol (ICMPv6)
         for the Internet Protocol Version 6 (IPv6) Specification
         - ICMPv6
         RFC 4340: Datagram Congestion Control Protocol";
    }

    identity code {
      base icmpv4;
      base icmpv6;
      description
        "Identity for ICMPv4 and ICMPv6 code condition capability";
      reference
        "RFC 792: Internet Control Message Protocol
         RFC 4443: Internet Control Message Protocol (ICMPv6)
         for the Internet Protocol Version 6 (IPv6) Specification
         - ICMPv6";
    }

    identity application-protocol {
      base protocol;
      description
        "Base identity for Application protocol. Note that a subset of
         application protocols (e.g., HTTP, HTTPS, FTP, POP3, and
         IMAP) are handled in this YANG module, rather than all
         the existing application protocols.";
    }

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

    identity https {
      base application-protocol;
      description
        "The identity for Hypertext Transfer Protocol version 1.1
         (HTTP/1.1) over TLS.";
      reference
```

```
         "draft-ietf-httpbis-semantics-19: HTTP Semantics
          draft-ietf-httpbis-messaging-19: HTTP/1.1";
     }

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

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

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

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

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

     identity smtp {
       base application-protocol;
```

```
      description
        "The identity for Simple Mail Transfer Protocol.";
      reference
        "RFC 5321: Simple Mail Transfer Protocol (SMTP)";
    }

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

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

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

    identity imaps {
      base application-protocol;
      description
        "The identity for Internet Message Access Protocol (IMAP) over
         TLS";
      reference
        "RFC 9051: Internet Message Access Protocol (IMAP) - Version
         4rev2
         RFC 2595: Using TLS with IMAP, POP3 and ACAP";
    }

    identity action {
      description
        "Base identity for action capability";
    }
```

```
identity log-action {
  base action;
  description
    "Base identity for log-action capability";
}

identity ingress-action {
  base action;
  description
    "Base identity for ingress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Section 7.2";
}

identity egress-action {
  base action;
  description
    "Base identity for egress-action capability";
  reference
    "RFC 8329: Framework for Interface to Network Security
     Functions - Section 7.2";
}

identity default-action {
  base action;
  description
    "Base identity for default-action capability";
}

identity rule-log {
  base log-action;
  description
    "Identity for rule log. Log the policy rule that has been
     triggered.";
}

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

identity pass {
```

```
      base ingress-action;
      base egress-action;
      base default-action;
      description
        "Identity for pass action capability. The pass action allows
         packet or flow to go through the NSF entering or exiting the
         internal network.";
    }

    identity drop {
      base ingress-action;
      base egress-action;
      base default-action;
      description
        "Identity for drop action capability. The drop action denies
         a packet to go through the NSF entering or exiting the
         internal network without sending any response back to the
         source.";
    }

    identity reject {
      base ingress-action;
      base egress-action;
      base default-action;
      description
        "Identity for reject action capability. 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
        "Identity for mirror action capability. The mirror action
         copies packet and send it to the monitoring entity while still
         allow the packet or flow to go through the NSF.";
    }

    identity rate-limit {
      base ingress-action;
```

```
      base egress-action;
      base default-action;
      description
        "Identity for rate limiting action capability. 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).";
    }

    identity invoke-signaling {
      base egress-action;
      description
        "Identity for invoke signaling action capability. 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
        "Identity for tunnel encapsulation action capability. 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
        "Identity for forwarding action capability. 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
        "Identity for transformation action capability. 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 and
         Network Address Translator (NAT)) 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
          RFC 3022: Traditional IP Network Address Translator
          (Traditional NAT)";
     }

     identity http-to-coap {
       base transformation;
       description
         "Identity for HTTP-to-CoAP transformation action capability.
          This indicates the support of HTTP-to-CoAP packet
          translation.";
       reference
         "RFC 8075: Guidelines for Mapping Implementations: HTTP to the
          Constrained Application Protocol (CoAP) - Translation between
          HTTP and CoAP.";
     }

     identity nat {
       base transformation;
       description
         "Identity for Network Address Translation (NAT) transformation
          action capability. This indicates the support of NAT for
          network address mapping.";
       reference
         "RFC 3022: Traditional IP Network Address Translator
          (Traditional NAT)";
     }

     identity resolution-strategy {
       description
         "Base identity for resolution strategy capability";
     }

     identity fmr {
       base resolution-strategy;
       description
         "Identity for First Matching Rule (FMR) resolution
          strategy capability";
     }

     identity lmr {
       base resolution-strategy;
       description
         "Identity for Last Matching Rule (LMR) resolution
          strategy capability";
     }
```

```
     identity pmre {
       base resolution-strategy;
       description
         "Identity for Prioritized Matching Rule with Errors (PMRE)
          resolution strategy capability";
     }

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

     identity advanced-nsf {
       description
         "Base identity for advanced Network Security Function (NSF)
          capability.";
     }

     identity content-security-control {
       base advanced-nsf;
       description
         "Base identity for content security control. Content security
          control is an NSF that evaluates a packet's payload such as
          Intrusion Prevention System (IPS), URL-Filtering, Antivirus,
          and VoIP/CN Filter.";
     }

     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
          or DDoS-mitigator.";
     }

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

     identity url-filtering {
       base content-security-control;
       description
         "Base identity for url filtering capability that 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
         "Base identity for antivirus capability to protect the network
          by detecting and removing viruses.";
     }

     identity voip-vocn-filtering {
       base content-security-control;
       description
         "Base identity for an advanced NSF for VoIP (Voice over
          Internet Protocol) and VoCN (Voice over Cellular Network,
          such as Voice over LTE or 5G) Security Service capability
          to filter the VoIP/VoCN packets or flows.";
       reference
         "RFC 3261: SIP: Session Initiation Protocol";
     }

     identity anti-ddos {
       base attack-mitigation-control;
       description
         "Base identity for advanced NSF Anti-DDoS Attack or DDoS
          Mitigator capability.";
     }

     identity packet-rate {
       base anti-ddos;
       description
         "Identity for advanced NSF Anti-DDoS detecting Packet Rate
          Capability where a packet rate is defined as the arrival rate
          of Packets toward a victim destination node.  The NSF with
          this capability can detect the incoming packet rate and create
          an alert if the rate exceeds the threshold.";

     }

     identity flow-rate {
       base anti-ddos;
       description
         "Identity for advanced NSF Anti-DDoS detecting Flow Rate
          Capability where a flow rate is defined as the arrival rate of
          flows towards a victim destination node.  The NSF with this
          capability can detect the incoming flow rate and create an
          alert if the rate exceeds the threshold.";
```

```
     }

     identity byte-rate {
       base anti-ddos;
       description
         "Identity for advanced NSF Anti-DDoS detecting Byte Rate
          Capability where a byte rate is defined as the arrival rate of
          Bytes toward a victim destination node.  The NSF with this
          capability can detect the incoming byte rate and create an
          alert if the rate exceeds the threshold.";
     }

     identity signature-set {
       base ips;
       description
         "Identity for the capability of IPS to set the signature.
          Signature is a set of rules to detect an intrusive activity.";
       reference
         "RFC 4766: Intrusion Detection Message Exchange Requirements -
          Section 2.2.13";
     }

     identity exception-signature {
       base ips;
       description
         "Identity for the capability of IPS to exclude signatures from
          detecting the intrusion.";
       reference
         "RFC 4766: Intrusion Detection Message Exchange Requirements -
          Section 2.2.13";
     }

     identity detect {
       base anti-virus;
       description
         "Identity for advanced NSF Antivirus capability to detect
          viruses using a security profile. The security profile is used
          to scan threats, such as virus, malware, and spyware. The NSF
          should be able to update the security profile.";
     }

     identity exception-files {
       base anti-virus;
       description
         "Identity for advanced NSF Antivirus capability to exclude a
          certain file type or name from detection.";
     }
```

```
     identity pre-defined {
       base url-filtering;
       description
         "Identity for pre-defined URL Database condition capability
          where URL database is a public database for URL filtering.";
     }

     identity user-defined {
       base url-filtering;
       description
         "Identity for user-defined URL Database condition capability
          that allows a user's manual addition of URLs for URL
          filtering.";
     }

     identity call-id {
       base voip-vocn-filtering;
       description
         "Identity for advanced NSF VoIP/VoCN Call Identifier (ID)
          capability.";
     }

     identity user-agent {
       base voip-vocn-filtering;
       description
         "Identity for advanced NSF VoIP/VoCN User Agent capability.";
     }

     /*
      *  Grouping
      */

     grouping nsf-capabilities {
       description
         "Network Security Function (NSF) Capabilities";
       reference
         "RFC 8329: Framework for Interface to Network Security
          Functions - I2NSF Flow Security Policy Structure.";

       leaf-list directional-capabilities {
         type identityref {
           base directional;
         }
         description
           "The capability of an NSF for handling directional traffic
            flow (i.e., unidirectional or bidirectional traffic flow).";
       }
```

```
       container event-capabilities {
         description
           "Capabilities of events.
            If a network security function has the event capabilities,
            the network security function supports rule execution
            according to system event and system alarm.";

         reference
           "RFC 8329: Framework for Interface to Network Security
            Functions - Section 7.
            draft-ietf-i2nsf-nsf-monitoring-data-model-19: I2NSF
            NSF Monitoring Interface YANG Data Model - System Alarm and
            System Events.";

         leaf-list system-event-capability {
           type identityref {
             base system-event;
           }
           description
             "System event capabilities";
         }

         leaf-list system-alarm-capability {
           type identityref {
             base system-alarm;
           }
           description
             "System alarm capabilities";
         }
       }

       container condition-capabilities {
         description
           "Conditions capabilities.";
         container generic-nsf-capabilities {
           description
             "Conditions capabilities.
              If a network security function has the condition
              capabilities, the network security function
              supports rule execution according to conditions of
              IPv4, IPv6, TCP, UDP, SCTP, DCCP, ICMP, or ICMPv6.";
           reference
             "RFC 768: User Datagram Protocol - UDP.
              RFC 791: Internet Protocol - IPv4.
              RFC 792: Internet Control Message Protocol - ICMP.
              RFC 4443: Internet Control Message Protocol (ICMPv6)
              for the Internet Protocol Version 6 (IPv6) Specification
              - ICMPv6.
```

```
              draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
              Transmission Protocol - SCTP.
              RFC 8200: Internet Protocol, Version 6 (IPv6)
              Specification - IPv6.
              RFC 8329: Framework for Interface to Network Security
              Functions - I2NSF Flow Security Policy Structure.
              draft-ietf-tcpm-rfc793bis-25: Transmission Control
              Protocol (TCP) Specification";

            leaf-list ethernet-capability {
              type identityref {
                base ethernet;
              }
              description
                "Media Access Control (MAC) capabilities";
              reference
                "IEEE 802.3: IEEE Standard for Ethernet";
            }

            leaf-list ipv4-capability {
              type identityref {
                base ipv4;
              }
              description
                "IPv4 packet capabilities";
              reference
                "RFC 791: Internet Protocol";
            }

            leaf-list ipv6-capability {
              type identityref {
                base ipv6;
              }
              description
                "IPv6 packet capabilities";
              reference
                "RFC 8200: Internet Protocol, Version 6 (IPv6)
                 Specification - IPv6";
            }

            leaf-list icmpv4-capability {
              type identityref {
                base icmpv4;
              }
              description
                "ICMPv4 packet capabilities";
              reference
                "RFC 792: Internet Control Message Protocol - ICMP";
```

```
          }

          leaf-list icmpv6-capability {
            type identityref {
              base icmpv6;
            }
            description
              "ICMPv6 packet capabilities";
            reference
              "RFC 4443: Internet Control Message Protocol (ICMPv6)
               for the Internet Protocol Version 6 (IPv6) Specification
               - ICMPv6";
          }

          leaf-list tcp-capability {
            type identityref {
              base tcp;
            }
            description
              "TCP packet capabilities";
            reference
              "draft-ietf-tcpm-rfc793bis-25: Transmission Control
               Protocol (TCP) Specification";
          }

          leaf-list udp-capability {
            type identityref {
              base udp;
            }
            description
              "UDP packet capabilities";
            reference
              "RFC 768: User Datagram Protocol - UDP";
          }

          leaf-list sctp-capability {
            type identityref {
              base sctp;
            }
            description
              "SCTP packet capabilities";
            reference
              "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
               Transmission Protocol - SCTP";
          }

          leaf-list dccp-capability {
            type identityref {
```

```
              base dccp;
            }
            description
              "DCCP packet capabilities";
            reference
              "RFC 4340: Datagram Congestion Control Protocol - DCCP";
          }
        }

        container advanced-nsf-capabilities {
          description
            "Advanced Network Security Function (NSF) capabilities,
             such as Anti-DDoS, IPS, and VoIP/VoCN.
             This container contains the leaf-lists of advanced
             NSF capabilities";

          leaf-list anti-ddos-capability {
            type identityref {
              base anti-ddos;
            }
            description
              "Anti-DDoS Attack capabilities";
          }

          leaf-list ips-capability {
            type identityref {
              base ips;
            }
            description
              "IPS capabilities";
          }

          leaf-list anti-virus-capability {
            type identityref {
              base anti-virus;
            }
            description
              "Antivirus capabilities";
          }

          leaf-list url-filtering-capability {
            type identityref {
              base url-filtering;
            }
            description
              "URL Filtering capabilities";
          }
```

```
          leaf-list voip-vocn-filtering-capability {
            type identityref {
              base voip-vocn-filtering;
           }
            description
              "VoIP/VoCN capabilities";
          }
        }

        container context-capabilities {
          description
            "Security context capabilities";

          leaf-list time-capabilities {
            type identityref {
              base time;
            }
            description
              "The capabilities for activating the policy within a
               specific time.";
          }

          leaf-list application-filter-capabilities{
            type identityref {
              base application-protocol;
            }
            description
              "Context capabilities based on the application protocol";
          }

          leaf-list device-type-capabilities {
            type identityref {
              base device-type;
            }
            description
              "Context capabilities based on the device attribute that
               can identify a device type
               (i.e., router, switch, pc, ios, or android).";
          }

          leaf-list user-condition-capabilities {
            type identityref {
              base user-condition;
            }
            description
              "Context capabilities based on user condition, such as
               user-id and user-name. The users can be collected into a
               user group (i.e., a group of users) and identified with
```

```
                 group-id or group-name. An NSF is aware of the IP
                 address of the user provided by a unified user
                 management system via network. Based on name-address
                 association, an NSF is able to enforce the security
                 functions over the given user (or user group)";
           }

           leaf-list geographic-capabilities {
             type identityref {
               base geographic-location;
             }
             description
               "Context condition capabilities based on the geographical
                location of the source or destination";
           }
         }
       }

       container action-capabilities {
         description
           "Action capabilities.
            If a network security function has the action capabilities,
            the network security function supports the attendant
            actions for policy rules.";

         leaf-list ingress-action-capability {
           type identityref {
             base ingress-action;
           }
           description
             "Ingress-action capabilities";
         }

         leaf-list egress-action-capability {
           type identityref {
             base egress-action;
           }
           description
             "Egress-action capabilities";
         }

         leaf-list log-action-capability {
           type identityref {
             base log-action;
           }
           description
             "Log-action capabilities";
         }
```

```
      }

      leaf-list resolution-strategy-capabilities {
        type identityref {
          base resolution-strategy;
        }
        description
          "Resolution strategy capabilities.
           The resolution strategies can be used to specify how
           to resolve conflicts that occur between the actions
           of the similar or different policy rules that are matched
           for the same packet and by particular NSF; note that a
           badly written policy rule may cause a conflict of actions
           with another similar policy rule.";
      }

      leaf-list default-action-capabilities {
        type identityref {
          base default-action;
        }
        description
          "Default action capabilities.
           A default action is used to execute I2NSF policy rules
           when no rule matches a packet. The default action is
           defined as pass, drop, reject, rate-limit, or mirror.";
      }
    }

    /*
     * Data nodes
     */

    list nsf {
      key "nsf-name";
      description
        "The list of Network Security Functions (NSFs)";
      leaf nsf-name {
        type string;
        mandatory true;
        description
          "The name of Network Security Function (NSF)";
      }
      uses nsf-capabilities;
    }
  }
  <CODE ENDS>
```

Figure 3: YANG Data Module of I2NSF Capability

7. IANA Considerations

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

ID: yang:ietf-i2nsf-capability
URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability
Registrant Contact: The IESG.
XML: N/A; the requested URI is an XML namespace.
Filename: [TBD-at-Registration]
Reference: [RFC-to-be]

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

Name: ietf-i2nsf-capability
Maintained by IANA? N
Namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability
Prefix: i2nsfcap
Module:
Reference: [RFC-to-be]

8. Privacy Considerations

This YANG module specifies the capabilities of NSFs. These capabilities are consistent with the diverse set of network security functions in common use in enterprise security operations. The configuration of the capabilities may entail privacy-sensitive information as explicitly outlined in Section 9. The NSFs implementing these capabilities may inspect, alter or drop user traffic; and be capable of attributing user traffic to individual users.

Due to the sensitivity of these capabilities, notice must be provided to and consent must be received from the users of the network. Additionally, the collected data and associated infrastructure must be secured to prevent the leakage or unauthorized disclosure of this private data.

9. Security Considerations

The YANG module specified in this document defines a data schema designed to be accessed through network management protocols such as NETCONF [RFC6241] or RESTCONF [RFC8040]. The lowest layer of NETCONF protocol layers MUST use Secure Shell (SSH) [RFC4254][RFC6242] as a secure transport layer. The lowest layer of RESTCONF protocol layers MUST use HTTP over Transport Layer Security (TLS) [RFC8446], that is, HTTPS as a secure transport layer.

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

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

* list nsf: An attacker could alter the security capabilities associated with an NSF in the database maintained by the security controller. Such changes could result in security functionality going unused due to the controller not having a record of it, and could also result in falsely claiming security capabilities that the controller would then attempt to use but would not actually be provided.

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

* list nsf: The leak of this node to an attacker could reveal the specific configuration of security controls to an attacker. An attacker can craft an attack path that avoids observation or mitigations by getting the information of available security capabilities in a victim network.

Some of the capability indicators (i.e., identities) defined in this document are highly sensitive and/or privileged operations that inherently require access to individuals' private data. These are subtrees and data nodes that are considered privacy-sensitive:

* url-filtering-capability: URLs themselves often contain sensitive information [CAPABILITY-URLS], and access to URLs typically comes hand-in-hand with access to request and response content, which is also often sensitive.

* voip-vocn-filtering-capability: The NSF that is able to filter VoIP/VoCN calls might identify certain individual identification.

* user-condition-capabilities: The capability uses a set of IP addresses mapped to users.

* geographic-capabilities: The IP address used in this capability can identify a user's geographical location.

It is noted that some private information is made accessible in this manner. Thus, the nodes/entities given access to this data MUST be tightly secured, monitored, and audited to prevent leakage or other unauthorized disclosure of private data. Refer to [RFC6973] for the description of privacy aspects that protocol designers (including YANG data model designers) should consider along with regular security and privacy analysis.

10. References

10.1. Normative References

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

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

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

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

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

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

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

[RFC2474] Nichols, K., Blake, S., Baker, F., and D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers", RFC 2474, DOI 10.17487/RFC2474, December 1998, <https://www.rfc-editor.org/info/rfc2474>.

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

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

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, DOI 10.17487/RFC3168, September 2001, <https://www.rfc-editor.org/info/rfc3168>.

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

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

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

[RFC4254] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Connection Protocol", RFC 4254, DOI 10.17487/RFC4254, January 2006, <https://www.rfc-editor.org/info/rfc4254>.

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

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

[RFC4766] Wood, M. and M. Erlinger, "Intrusion Detection Message Exchange Requirements", RFC 4766, DOI 10.17487/RFC4766, March 2007, <https://www.rfc-editor.org/info/rfc4766>.

[RFC5103] Trammell, B. and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, DOI 10.17487/RFC5103, January 2008, <https://www.rfc-editor.org/info/rfc5103>.

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

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

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

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

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

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

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC6691] Borman, D., "TCP Options and Maximum Segment Size (MSS)", RFC 6691, DOI 10.17487/RFC6691, July 2012, <https://www.rfc-editor.org/info/rfc6691>.

[RFC6864] Touch, J., "Updated Specification of the IPv4 ID Field", RFC 6864, DOI 10.17487/RFC6864, February 2013, <https://www.rfc-editor.org/info/rfc6864>.

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

[RFC7323] Borman, D., Braden, B., Jacobson, V., and R. Scheffenegger, Ed., "TCP Extensions for High Performance", RFC 7323, DOI 10.17487/RFC7323, September 2014, <https://www.rfc-editor.org/info/rfc7323>.

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

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

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

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

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

[RFC8311] Black, D., "Relaxing Restrictions on Explicit Congestion Notification (ECN) Experimentation", RFC 8311, DOI 10.17487/RFC8311, January 2018, <https://www.rfc-editor.org/info/rfc8311>.

[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>.

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

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

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

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

[RFC8805] Kline, E., Duleba, K., Szamonek, Z., Moser, S., and W. Kumari, "A Format for Self-Published IP Geolocation Feeds", RFC 8805, DOI 10.17487/RFC8805, August 2020, <https://www.rfc-editor.org/info/rfc8805>.

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

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

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

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

[I-D.ietf-i2nsf-nsf-facing-interface-dm] Kim, J. T., Jeong, J. P., Park, 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-27, 14 May 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-facing-interface-dm-27.txt>.

[I-D.ietf-i2nsf-nsf-monitoring-data-model] Jeong, J. (., 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-18, 19 April 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-monitoring-data-model-18.txt>.

[I-D.ietf-i2nsf-registration-interface-dm] Hyun, S., Jeong, J. (., Roh, T., Wi, S., and J. Park, "I2NSF Registration Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-registration-interface-dm-16, 13 April 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-registration-interface-dm-16.txt>.

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

[I-D.ietf-tcpm-accurate-ecn]
Briscoe, B., Kühlewind, M., and R. Scheffenegger, "More Accurate ECN Feedback in TCP", Work in Progress, Internet-Draft, draft-ietf-tcpm-accurate-ecn-18, 22 March 2022, <https://www.ietf.org/archive/id/draft-ietf-tcpm-accurate-ecn-18.txt>.

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

[I-D.ietf-tsvwg-udp-options]
Touch, J., "Transport Options for UDP", Work in Progress, Internet-Draft, draft-ietf-tsvwg-udp-options-18, 26 March 2022, <https://www.ietf.org/archive/id/draft-ietf-tsvwg-udp-options-18.txt>.

10.2. Informative References

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <https://www.rfc-editor.org/info/rfc6973>.

[RFC8192] Hares, S., Lopez, D., Zarny, M., Jacquenet, C., Kumar, R., and J. Jeong, "Interface to Network Security Functions (I2NSF): Problem Statement and Use Cases", RFC 8192, DOI 10.17487/RFC8192, July 2017, <https://www.rfc-editor.org/info/rfc8192>.

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

[IANA-Protocol-Numbers]
"Assigned Internet Protocol Numbers", Available: https://www.iana.org/assignments/protocol-numbers/protocol-numbers.xhtml, September 2020.

[IEEE802.3-2018]
Committee, I. S., "IEEE 802.3-2018 - IEEE Standard for Ethernet", August 2018, <https://ieeexplore.ieee.org/document/8457469>.

[Alshaer] Shaer, Al., Hamed, E., and H. Hamed, "Modeling and management of firewall policies", 2004.

[Hirschman]
Hirschman, L. and R. Gaizauskas, "Natural Language Question Answering: The View from Here", Natural Language Engineering 7:4, pgs 275-300, Cambridge University Press , November 2001.

[Hohpe] Hohpe, G. and B. Woolf, "Enterprise Integration Patterns", ISBN 0-32-120068-3 , 2003.

[Martin] Martin, R.C., "Agile Software Development, Principles, Patterns, and Practices", Prentice-Hall , ISBN: 0-13-597444-5 , 2002.

[OODMP] "https://www.oodesign.com/mediator-pattern.html".

[OODOP] "https://www.oodesign.com/observer-pattern.html".

[OODSRP] "https://www.oodesign.com/single-responsibility-principle.html".

[CAPABILITY-URLS]
Tennison, J., "Good Practices for Capability URLs", October 2014, <https://www.w3.org/2001/tag/doc/capability-urls/>.

Appendix A. Configuration Examples

This section shows configuration examples of "ietf-i2nsf-capability" module for capabilities registration of general firewall.

A.1. Example 1: Registration for the Capabilities of a General Firewall

This section shows a configuration example for the capabilities registration of a general firewall in either an IPv4 network or an IPv6 network.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>general_firewall</nsf-name>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv4-capability>next-header</ipv4-capability>
   <ipv4-capability>flow-direction</ipv4-capability>
   <ipv4-capability>source-address</ipv4-capability>
   <ipv4-capability>destination-address</ipv4-capability>
   <tcp-capability>source-port-number</tcp-capability>
   <tcp-capability>destination-port-number</tcp-capability>
   <udp-capability>source-port-number</udp-capability>
   <udp-capability>destination-port-number</udp-capability>
  </generic-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 4: Configuration XML for the Capabilities Registration of a General Firewall in an IPv4 Network

Figure 4 shows the configuration XML for the capabilities registration of a general firewall as an NSF in an IPv4 network. Its capabilities are as follows.

1. The name of the NSF is general_firewall.

2. The NSF can inspect the IPv4 protocol header field, flow direction, source address(es), and destination address(es)

3. The NSF can inspect the port number(s) and flow direction for the transport layer protocol, i.e., TCP and UDP.

4. The NSF can control whether the packets are allowed to pass, drop, or mirror.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>general_firewall</nsf-name>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv6-capability>next-header</ipv6-capability>
   <ipv6-capability>flow-direction</ipv6-capability>
   <ipv6-capability>source-address</ipv6-capability>
   <ipv6-capability>destination-address</ipv6-capability>
   <tcp-capability>source-port-number</tcp-capability>
   <tcp-capability>destination-port-number</tcp-capability>
   <udp-capability>source-port-number</udp-capability>
   <udp-capability>destination-port-number</udp-capability>
  </generic-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 5: Configuration XML for the Capabilities Registration of a General Firewall in an IPv6 Network

In addition, Figure 5 shows the configuration XML for the capabilities registration of a general firewall as an NSF in an IPv6 network. Its capabilities are as follows.

1. The name of the NSF is general_firewall.

2. The NSF can inspect IPv6 next header, flow direction, source address(es), and destination address(es)

3. The NSF can inspect the port number(s) and flow direction for the transport layer protocol, i.e., TCP and UDP.

4. The NSF can control whether the packets are allowed to pass, drop, or mirror.

A.2. Example 2: Registration for the Capabilities of a Time-based Firewall

This section shows a configuration example for the capabilities registration of a time-based firewall in either an IPv4 network or an IPv6 network.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>time_based_firewall</nsf-name>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv4-capability>next-header</ipv4-capability>
   <ipv4-capability>flow-direction</ipv4-capability>
   <ipv4-capability>source-address</ipv4-capability>
   <ipv4-capability>destination-address</ipv4-capability>
   <context-capabilities>
    <time-capabilities>absolute-time</time-capabilities>
    <time-capabilities>periodic-time</time-capabilities>
   </context-capabilities>
  </generic-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 6: Configuration XML for the Capabilities Registration of a Time-based Firewall in an IPv4 Network

Figure 6 shows the configuration XML for the capabilities registration of a time-based firewall as an NSF in an IPv4 network. Its capabilities are as follows.

1. The name of the NSF is time_based_firewall.

2. The NSF can execute the security policy rule according to absolute time and periodic time.

3. The NSF can inspect the IPv4 protocol header field, flow direction, source address(es), and destination address(es).

4. The NSF can control whether the packets are allowed to pass, drop, or mirror.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>time_based_firewall</nsf-name>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv6-capability>next-header</ipv6-capability>
   <ipv6-capability>flow-direction</ipv6-capability>
   <ipv6-capability>source-address</ipv6-capability>
   <ipv6-capability>destination-address</ipv6-capability>
   <context-capabilities>
    <time-capabilities>absolute-time</time-capabilities>
    <time-capabilities>periodic-time</time-capabilities>
   </context-capabilities>
  </generic-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 7: Configuration XML for the Capabilities Registration of a Time-based Firewall in an IPv6 Network

In addition, Figure 7 shows the configuration XML for the capabilities registration of a time-based firewall as an NSF in an IPv6 network. Its capabilities are as follows.

1. The name of the NSF is time_based_firewall.

2. The NSF can execute the security policy rule according to absolute time and periodic time.

3. The NSF can inspect the IPv6 protocol header field, flow direction, source address(es), and destination address(es).

4. The NSF can control whether the packets are allowed to pass, drop, or mirror.

A.3. Example 3: Registration for the Capabilities of a Web Filter

This section shows a configuration example for the capabilities registration of a web filter.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>web_filter</nsf-name>
 <condition-capabilities>
  <advanced-nsf-capabilities>
   <url-filtering-capability>user-defined</url-filtering-capability>
  </advanced-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 8: Configuration XML for the Capabilities Registration of a Web Filter

Figure 8 shows the configuration XML for the capabilities registration of a web filter as an NSF. Its capabilities are as follows.

1. The name of the NSF is web_filter.

2. The NSF can inspect a URL matched from a user-defined URL. User can specify their own URL.

3. The NSF can control whether the packets are allowed to pass, drop, or mirror.

4. Overall, the NSF can compare the URL of a packet to a user-defined database. The matched packet can be passed, dropped, or mirrored.

A.4. Example 4: Registration for the Capabilities of a VoIP/VoCN Filter

This section shows a configuration example for the capabilities registration of a VoIP/VoCN filter.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>voip_vocn_filter</nsf-name>
 <condition-capabilities>
  <advanced-nsf-capabilities>
   <voip-vocn-filtering-capability>
    call-id
   </voip-vocn-filtering-capability>
  </advanced-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 9: Configuration XML for the Capabilities Registration of a VoIP/VoCN Filter

Figure 9 shows the configuration XML for the capabilities registration of a VoIP/VoCN filter as an NSF. Its capabilities are as follows.

1. The name of the NSF is voip_vocn_filter.

2. The NSF can inspect a voice call id for VoIP/VoCN packets.

3. The NSF can control whether the packets are allowed to pass, drop, or mirror.

A.5. Example 5: Registration for the Capabilities of an HTTP and HTTPS Flood Mitigator

This section shows a configuration example for the capabilities registration of a HTTP and HTTPS flood mitigator.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>DDoS_mitigator</nsf-name>
 <condition-capabilities>
  <advanced-nsf-capabilities>
   <anti-ddos-capability>packet-rate</anti-ddos-capability>
   <anti-ddos-capability>byte-rate</anti-ddos-capability>
   <anti-ddos-capability>flow-rate</anti-ddos-capability>
  </advanced-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>mirror</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>mirror</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 10: Configuration XML for the Capabilities Registration of a HTTP and HTTPS Flood Mitigator

Figure 10 shows the configuration XML for the capabilities registration of a HTTP and HTTPS flood mitigator as an NSF. Its capabilities are as follows.

1. The name of the NSF is DDoS_mitigator.

2. The NSF can detect the amount of packet, flow, and byte rate in the network for potential DDoS Attack.

3. The NSF can control whether the packets are allowed to pass, drop, or mirror.

Appendix B. 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 experts: Roman Danyliw, Acee Lindem, Paul Wouters (SecDir), Michael Scharf (TSVART), Dan Romascanu (GenART), 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 grant funded by the MSIT (2020-0-00395, Standard Development of Blockchain based Network Management Automation Technology).

Appendix C. 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

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

Cataldo Basile - Politecnico di Torino, Corso Duca degli Abruzzi, 34, Torino, 10129, Italy, EMail: cataldo.basile@polito.it

John Strassner - Huawei, 2330 Central Expressway, Santa Clara, CA 95050, USA, EMail: John.sc.Strassner@huawei.com

Diego R. Lopez - Telefonica I+D, Zurbaran, 12, Madrid, 28010, Spain, Email: diego.r.lopez@telefonica.com

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

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

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

Jung-Soo Park - Electronics and Telecommunications Research Institute, 218 Gajeong-Ro, Yuseong-Gu, Daejeon, 34129, Republic of Korea, EMail: pjs@etri.re.kr

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

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

Appendix D. Changes from draft-ietf-i2nsf-capability-data-model-31

The following changes are made from draft-ietf-i2nsf-capability-data-model-31:

* The YANG module's prefix is updated from 'nsfcap' to 'i2nsfcap'.

Authors' Addresses

Susan Hares (editor)
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America
Phone: +1-734-604-0332
Email: shares@ndzh.com

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

Jinyong Tim Kim
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

Robert Moskowitz
HTT Consulting
Oak Park, MI
United States of America
Phone: +1-248-968-9809
Email: rgm@htt-consult.com

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen
Guangdong 518129,
China
Email: linqiushi@huawei.com

I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 16 November 2023

J. Jeong, Ed.
C. Chung
Sungkyunkwan University
T. Ahn
Korea Telecom
R. Kumar
Juniper Networks
S. Hares
Huawei

15 May 2023

I2NSF Consumer-Facing Interface YANG Data Model
draft-ietf-i2nsf-consumer-facing-interface-dm-31

Abstract

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

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

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

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

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

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

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

Data models are defined at a lower level of abstraction and provide many details. They provide details about the implementation of a protocol's specification, e.g., rules that explain how to map managed objects onto lower-level protocol constructs.

The efficient and flexible provisioning of network functions by a Network Functions Virtualization (NFV) system supports rapid deployment of newly developed functions. As practical applications, Network Security Functions (NSFs), such as firewall, Intrusion Detection System (IDS)/Intrusion Prevention System (IPS), and attack mitigation, can also be provided as Virtual Network Functions (VNFs) in the NFV system. By the efficient virtualization technology, these VNFs might be automatically provisioned and dynamically migrated based on real-time security requirements. This document presents a YANG data model to implement security functions based on NFV.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terminology described in [RFC8329].

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

3. YANG Tree Diagram of Policy

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

Name: This field identifies the name of this object.

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

Priority-usage: This field represents the type of the priority used in the policy. Two types are defined in this module, i.e., 'priority-by-order' and 'priority-by-number'. The 'priority-by-order' indicates that the sequence of the rules to be executed follows the input order by user. The 'priority-by-number' indicates that the sequence of the rules to be executed follows the priority values in the rules, where a higher priority value means a higher priority.

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

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

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

Figure 2: Policy YANG Data Tree

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

Name: This field identifies the name of this object.

Priority: This field identifies the priority of the rule. This

field can be given when the policy's 'priority-usage' is priority-by-number.

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

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

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

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

Figure 3: Rule YANG Data Tree

3.1. Event Sub-model

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

System-event (also called alert): is defined as a warning about any

changes of configuration, any access violation, the information of sessions and traffic flows.

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

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

Figure 4: Event Sub-model YANG Data Tree

3.2. Condition Sub-model

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

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

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

anti-virus: This field represents the configuration for an Antivirus

service. A specific security profile can be added to Security Controller in order to update the configuration of the Antivirus service. Also, either a filename or path for such a profile can be configured for the Antivirus service.

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

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

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

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

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

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

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

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

```
|  |  |     +--rw id      uint32
|  |  |     +--rw name?   string
|  |  +--rw geographic-location
|  |     +--rw source
|  |     |  +--rw country?   -> /endpoint-groups/location-group/country
|  |     |  +--rw region?    -> /endpoint-groups/location-group/region
|  |     |  +--rw city?      -> /endpoint-groups/location-group/city
|  |     +--rw destination
|  |        +--rw country?   -> /endpoint-groups/location-group/country
|  |        +--rw region?    -> /endpoint-groups/location-group/region
|  |        +--rw city?      -> /endpoint-groups/location-group/city
|  +--rw threat-feed
|     +--rw name*   -> /threat-prevention/threat-feed-list/name
```

Figure 5: Condition Sub-model YANG Data Tree

3.3. Action Sub-model

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

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

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

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

Figure 6: Action Sub-model YANG Data Tree

4. YANG Tree Diagram of Policy Endpoint Groups

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

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

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

Figure 7: Endpoint Group Diagram

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

Figure 8: Endpoint Group YANG Data Tree

4.1. User-Group

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

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

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

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

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

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

Figure 9: User-Group YANG Data Tree

4.2. Device-Group

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

Name: This field identifies the name of this object.

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

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

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

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

Figure 10: Device-Group YANG Data Tree

4.3. Location-Group

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

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

Region: This field represents the region code conforming to ISO

3166-2. Examples include 'ID-RI' for Riau province of Indonesia and 'NG-RI' for the Rivers province in Nigeria.

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

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

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

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

Figure 11: Location-Group YANG Data Tree

4.4. URL-Group

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

Name: This field identifies the name of this object.

URL: This field represents the URL.

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

Figure 12: URL-Group YANG Data Tree

4.5. Voice-Group

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

Name: This field identifies the name of this object.

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

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

Figure 13: Voice-Group YANG Data Tree

5. YANG Tree Diagram of Threat Prevention

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

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

Figure 14: Threat Prevention Diagram

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

Figure 15: Threat Prevention YANG Data Tree

5.1. Threat Feed

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

Name: This field identifies the name of this object.

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

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

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

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

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

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

Figure 16: Threat Feed YANG Data Tree

5.2. Payload Content

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

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

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

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

Depth: This field specifies how far a packet should be searched

for the specified content pattern defined in the content field. If this field is undefined, then the content pattern should be searched within the whole payload.

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

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

Figure 17: Payload Content in YANG Data Tree

6. YANG Data Model of Consumer-Facing Interface

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

The semantics of the data model is aligned with the information model of the Consumer-Facing Interface. This data model is designed to support the I2NSF framework that can be extended according to the security needs. In other words, the model design is independent of the content and meaning of specific policies as well as the implementation approach.

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

6.1. YANG Module of Consumer-Facing Interface

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

```
<CODE BEGINS> file "ietf-i2nsf-cons-facing-interface@2023-05-15.yang"
module ietf-i2nsf-cons-facing-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-cons-facing-interface";
  prefix
    i2nsfcfi;

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

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

  import ietf-i2nsf-monitoring-interface {
    prefix i2nsfmi;
    reference
```

```
         "draft-ietf-i2nsf-nsf-monitoring-data-model-20";
         // RFC Ed.: replace with an actual RFC number and remove
         // this note.
     }

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

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

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

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

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

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
        'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
        'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this
        document are to be interpreted as described in BCP 14
        (RFC 2119) (RFC 8174) when, and only when, they appear
        in all capitals, as shown here.

        Copyright (c) 2023 IETF Trust and the persons identified as
        authors of the code. All rights reserved.

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

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

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

     revision "2023-05-15" {
       description "Initial revision.";
```

```
      reference
        "RFC XXXX: I2NSF Consumer-Facing Interface YANG Data Model";

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

    identity 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 user's input 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 priority number or value of the configuration. The higher
         the value is, the higher the priority is.";
    }

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

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

    identity lmr {
```

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

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

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

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

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

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

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

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

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

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

     identity reject {
       base ingress-action;
```

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

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

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

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

    identity tunnel-encapsulation {
```

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

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

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

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

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

    identity session-log {
      base log-action;
      description
        "A session is a connection (i.e., traffic flow) of a data plane
         that includes source and destination information of IP
```

```
          addresses and transport port numbers with the protocol used.
          Log the session that triggered a policy rule.";
     }

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

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

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

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

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

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

     identity router-solicitation {
       base icmp-message;
       description
         "Identity for 'Router Solicitation' ICMP message type 10 in
          ICMPv4 or type 135 in ICMPv6.";
     }

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

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

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

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

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

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

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

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

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

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

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

     identity multiple-interfaces-satisfy-query {
       base extended-echo-reply;
       description
```

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

    identity ioc-format {
      description
        "This represents the base identity for the format of the
         Indicators of Compromise (IOC).";
    }

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

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

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

    identity iodef {
      base ioc-format;
      description
        "This represents the Incident Object Description Exchange Format
         (IODEF) format.";
      reference
```

```
        "RFC 8727: JSON Binding of the Incident Object Description
         Exchange Format";
    }

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

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

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

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

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

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

    identity iot-device {
      base device-type;
```

```
      description
        "Identity for Internet of Things (IoT) devices
         such as sensors, actuators, and low-power
         low-capacity computing devices.";
    }

    identity ot {
      base device-type;
      description
        "Identity for Operational Technology (OT) devices (also
         known as industrial control systems) that interact
         with the physical environment and detect or cause direct
         change through the monitoring and control of devices,
         processes, and events such as programmable logic
         controllers (PLCs), digital oscilloscopes, building
         management systems (BMS), and fire control systems.";
    }

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

   /*
    * Typedefs
    */

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

    typedef day {
      type enumeration {
        enum monday {
          description
            "This represents Monday.";
        }
        enum tuesday {
          description
```

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

   /*
    * Groupings
    */

    grouping ip-address-info {
      description
        "There are two types to configure a security policy
         for an IP address, such as IPv4 address and IPv6 address.";
      choice match-type {
        description
          "User can choose between IPv4 and IPv6.";
        case ipv4 {
          choice ipv4-range-or-prefix {
            description
              "User can choose between IPv4 address range and
               prefix type.";
            case prefix {
              leaf-list ipv4-prefix {
                type inet:ipv4-prefix;
                description
                  "The IPv4 addresses in a prefix type.";
```

```
              }
            }
            case range {
              list range-ipv4-address {
                key "start end";
                leaf start {
                  type inet:ipv4-address-no-zone;
                  mandatory true;
                  description
                    "A start IPv4 address for a range match.";
                }
                leaf end {
                  type inet:ipv4-address-no-zone;
                  mandatory true;
                  description
                    "An end IPv4 address for a range match.";
                }
                description
                  "A range match for IPv4 addresses is provided.
                   The ranges are inclusive, i.e., the range values
                   include the value of 'start' and 'end'.
                   Note that the start IPv4 address must be lower than
                   the end IPv4 address.";
              }
            }
          }
        }
        case ipv6 {
          choice ipv6-range-or-prefix {
            description
              "User can choose between IPv6 address range and
               prefix type.";
            case prefix {
              leaf-list ipv6-prefix {
                type inet:ipv6-prefix;
                description
                  "The IPv6 addresses in a prefix type.";
              }
            }
            case range {
              list range-ipv6-address {
                key "start end";
                leaf start {
                  type inet:ipv6-address-no-zone;
                  mandatory true;
                  description
                    "A start IPv6 address for a range match.";
                }
```

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

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

     grouping device-group {
       description
```

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

    grouping location-group {
      description
        "This group represents location-group information to map
         IPv4 or IPv6 address to the geographical location.";
      leaf country {
        type string {
          length "2";
          pattern "[a-zA-Z]{2}";
        }
        description
          "This represents the 2-letter ISO country code conforming to
           ISO3166-1 alpha 2. Examples include 'US' for United States,
           'JP' for Japan, and 'PL' for Poland.";
        reference
          "ISO 3166-1: Decoding table alpha-2 country code";
      }
      leaf region {
        type string {
          length "5..6";
          pattern "[a-zA-Z]{2}-[a-zA-Z0-9]{2,3}";
        }
        description
```

```
          "This represents the ISO region code conforming to ISO 3166-2.
           Examples include 'ID-RI' for Riau province of Indonesia and
           'NG-RI' for the Rivers province in Nigeria.";
        reference
          "ISO 3166-2: 3166-2 subdivision code";
      }
      leaf city {
        type string;
        description
          "This represents the city of a region in English. Examples
           include 'Dublin', 'New York', and 'Sao Paulo'.";
      }
      uses ip-address-info{
        refine match-type{
          mandatory true;
          description
            "This represents the IP addresses of a location-group.";
        }
      }
    }

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

    list i2nsf-cfi-policy {
      key "name";
      description
        "This is a security policy list.  Each policy in the list
         contains a list of security policy rules, and is a policy
         instance to have the information of where and when a policy
         needs to be applied.";
      leaf name {
        type string;
        description
          "The name which identifies the policy.";
      }
      leaf language {
        type string {
          pattern '((([A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3})'
                + '{0,2})?)|[A-Za-z]{4}|[A-Za-z]{5,8})(-[A-Za-z]{4})?'
                + '(-([A-Za-z]{2}|[0-9]{3}))?(-([A-Za-z0-9]{5,8}'
                + '|([0-9][A-Za-z0-9]{3})))*(-[0-9A-WYZa-wyz]'
                + '(-([A-Za-z0-9]{2,8}))+)*(-[Xx](-([A-Za-z0-9]'
                + '{1,8}))+)?|[Xx](-([A-Za-z0-9]{1,8}))+|'
                + '(([Ee][Nn]-[Gg][Bb]-[Oo][Ee][Dd]|[Ii]-'
```

```
                + '[Aa][Mm][Ii]|[Ii]-[Bb][Nn][Nn]|[Ii]-'
                + '[Dd][Ee][Ff][Aa][Uu][Ll][Tt]|[Ii]-'
                + '[Ee][Nn][Oo][Cc][Hh][Ii][Aa][Nn]'
                + '|[Ii]-[Hh][Aa][Kk]|'
                + '[Ii]-[Kk][Ll][Ii][Nn][Gg][Oo][Nn]|'
                + '[Ii]-[Ll][Uu][Xx]|[Ii]-[Mm][Ii][Nn][Gg][Oo]|'
                + '[Ii]-[Nn][Aa][Vv][Aa][Jj][Oo]|[Ii]-[Pp][Ww][Nn]|'
                + '[Ii]-[Tt][Aa][Oo]|[Ii]-[Tt][Aa][Yy]|'
                + '[Ii]-[Tt][Ss][Uu]|[Ss][Gg][Nn]-[Bb][Ee]-[Ff][Rr]|'
                + '[Ss][Gg][Nn]-[Bb][Ee]-[Nn][Ll]|[Ss][Gg][Nn]-'
                + '[Cc][Hh]-[Dd][Ee])|([Aa][Rr][Tt]-'
                + '[Ll][Oo][Jj][Bb][Aa][Nn]|[Cc][Ee][Ll]-'
                + '[Gg][Aa][Uu][Ll][Ii][Ss][Hh]|'
                + '[Nn][Oo]-[Bb][Oo][Kk]|[Nn][Oo]-'
                + '[Nn][Yy][Nn]|[Zz][Hh]-[Gg][Uu][Oo][Yy][Uu]|'
                + '[Zz][Hh]-[Hh][Aa][Kk][Kk][Aa]|[Zz][Hh]-'
                + '[Mm][Ii][Nn]|[Zz][Hh]-[Mm][Ii][Nn]-'
                + '[Nn][Aa][Nn]|[Zz][Hh]-[Xx][Ii][Aa][Nn][Gg])))';
        }
        default "en-US";
        description
          "The value in this field indicates the language tag
           used for all of the 'leaf description' described in the
           'i2nsf-cfi-policy'.

           The attribute is encoded following the rules in Section 2.1
           in RFC 5646. The default language tag is 'en-US'.";
        reference
          "RFC 5646: Tags for Identifying Languages";
      }
      leaf 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";
      }
      list rules {
        key "name";

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

        leaf priority {
          when "derived-from-or-self(../../priority-usage,"
             + "'priority-by-number')";
          type uint8;
          description
            "The priority of the rule to indicate the order of the rules
             to be matched.  A higher value means a higher priority.
             The packet or flow will be matched with the rule with
             the highest priority value first and continues to a lower
             priority value.  Once a rule matches the packet or flow,
             the NSF should execute the rule and terminate the matching
             process.  If multiple rules have an equal priority, the
             actual order is undefined.  The handling of the selection
             of those rules depends on the implementer, e.g.,
             an alphabetical order of the rules' names or a random rule
             selection.";
        }

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

```
          }

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

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

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

            leaf transport-layer-protocol {
              type identityref {
```

```
                base i2nsfmi:transport-protocol;
              }
              description
                "The transport-layer protocol to be matched.";
            }

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

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

              leaf-list message {
                type identityref {
                  base icmp-message;
                }
                description
                  "The security policy rule according to
                   ICMP message. The type is representing the
```

```
                 ICMP message corresponds to the ICMP type and
                 code.";
               reference
                 "RFC  792: Internet Control Message Protocol
                 RFC 8335: PROBE: A Utility for Probing Interfaces
                 IANA: Internet Control Message Protocol (ICMP)
                 Parameters
                 IANA: Internet Control Message Protocol version 6
                 (ICMPv6) Parameters";
             }
           }
         }

         container ddos {
           description
             "A condition for a DDoS attack.";
           container rate-limit {
             description
               "This describes the rate-limit.";
             leaf packet-rate-threshold {
               type uint64;
               units "pps";
               description
                 "This is a trigger value for a rate limit of packet
                  rate in packets per second (pps) for a
                  DDoS-attack mitigation.";
             }
             leaf byte-rate-threshold {
               type uint64;
               units "Bps";
               description
                 "This is a trigger value for a rate limit of byte
                  rate in bytes per second (Bps) for a DDoS-attack
                  mitigation.";
             }
             leaf flow-rate-threshold {
               type uint64;
               description
                 "This is a trigger value for a rate limit of flow
                  creating request rate (e.g., new TCP connection
                  establishment) in flows per second for a DDoS-attack
                  mitigation.";
             }
           }
         }

         container anti-virus {
           description
```

```
              "A condition for Antivirus service";

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

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

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

          container url-category {
            description
              "Condition for url category.";
            leaf url-name {
              type leafref {
                path "/endpoint-groups/url-group/name";
              }
              description
                "This is description for the condition of a URL's
```

```
                 category such as SNS sites, game sites, e-commerce
                 sites, company sites, and university sites.";
           }
         }

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

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

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

           leaf-list user-agent {
```

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

          container context {
            description
              "Condition for matching the context of the packet, such
               as geographic location, time, packet direction.";
            container time {
              description
                "The time when a security policy rule should be
                 applied.";
              leaf start-date-time {
                type yang:date-and-time;
                description
                  "This is the start date and time for a security
                   policy rule.";
              }
              leaf end-date-time {
                type yang:date-and-time;
                description
                  "This is the end date and time for a security policy
                   rule.  The policy rule will stop working after the
                   specified end date and time.";
              }
              container period {
                when
                  "../frequency!='only-once'";
                description
                  "This represents the repetition time.";
                leaf start-time {
                  type time;
                  description
                    "This is a period's start time for an event.";
                }
                leaf end-time {
                  type time;
                  description
                    "This is a period's end time for an event.";
                }
                leaf-list day {
                  when
```

```
                    "../../frequency='weekly'";
                  type day;
                  min-elements 1;
                  description
                    "This represents the repeated day of every week
                     (e.g., Monday and Tuesday).  More than one day can
                     be specified.";
                }
                leaf-list date {
                  when
                    "../../frequency='monthly'";
                  type int8 {
                    range "1..31";
                  }
                  min-elements 1;
                  description
                    "This represents the repeated day of every month.
                     More than one date can be specified.";
                }
                list month {
                  when
                    "../../frequency='yearly'";

                  key "start end";

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

```
                 If one specific month and date is needed, then set
                 both start and end to the same value.

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

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

```
            }

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

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

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

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

            container geographic-location {
              description
                "A condition for a location-based connection.";
              container source {
                leaf country {
                  type leafref {
                    path "/endpoint-groups/location-group/country";
                  }
                  description
                    "The name of the country in the 2-letter ISO country
                     code conforming to ISO3166-1 alpha-2.";
                  reference
                    "ISO 3166-1: Decoding table alpha-2 country code";
                }
                leaf region {
                  type leafref {
                    path "/endpoint-groups/location-group/region";
                  }
                  description
                    "The region code conforming to ISO 3166-2.";
                  reference
                    "ISO 3166-2: 3166-2 subdivision code.";
                }
```

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

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

        container action {
          description
            "This is the action container.";
          container primary-action {
            description
              "This represents primary actions (e.g., ingress and
               egress actions) to be applied to a condition.
               If this is not set, it cannot support the primary
               actions.";
            leaf action {
              type identityref {
                base primary-action;
              }
              mandatory true;
              description
                "Ingress actions: pass, drop, reject, rate-limit,
                 and mirror.
                 Egress actions: pass, drop, reject, rate-limit,
                 mirror, invoke-signaling, tunnel-encapsulation,
                 forward, and transform.";
            }
            leaf limit {
              when "../action = 'i2nsfcfi:rate-limit'" {
                description
                  "Rate-limit is valid only when rate-limit action is
                   used.";
              }
              type decimal64 {
                fraction-digits 2;
              }
              units "bytes per second";
              description
                "Specifies how to rate-limit the traffic.";
            }
          }
```

```
          container secondary-action {
            description
              "This represents secondary actions (e.g., log and syslog)
               to be applied if they are needed.  If this is not set,
               it cannot support the secondary actions.";
            leaf log-action {
              type identityref {
                base secondary-action;
              }
              description
                "Log action: rule log and session log.";
            }
          }
        }
      }

      container endpoint-groups {
        description
          "A logical entity in a business environment, where a security
           policy is to be applied.";
        list user-group{
          uses user-group;
          key "name";
          description
            "This represents a user group.";
        }
        list device-group {
          key "name";
          uses device-group;
          description
            "This represents a device group.";
        }
        list location-group{
          key "country region city";
          uses location-group;
          description
            "This represents a location group.";
        }
        list url-group {
          key "name";
          description
            "This describes the list of URL.";
          leaf name {
            type string;
            description
              "This is the name of URL group, e.g., SNS sites,
               gaming sites, and e-commerce sites.";
```

```
        }
        leaf-list url {
          type inet:uri;
          description
            "Specifies the URL to be added into the group.";
          reference
            "RFC 3986: Uniform Resource Identifier (URI): Generic
             Syntax";
        }
      }
      list voice-group {
        key "name";
        description
          "This describes the list of Voice ID.";
        leaf name {
          type string;
          description
            "This is the name of the voice group.";
        }
        leaf-list sip-id {
          type inet:uri;
          description
            "Specifies the logical identity of the SIP user written in
             SIP URI scheme.";
          reference
            "RFC3261: SIP: Session Initiation Protocol
                      - Section 19.1.1 (SIP and SIPS URI Components)";
        }
      }
    }

    container threat-prevention {
      description
        "The container for threat-prevention.";
      list threat-feed-list {
        key "name";
        description
          "There can be a single or multiple number of
           threat-feeds.";
        leaf name {
          type string;
          description
            "This represents the name of the threat-feed.";
        }
        leaf-list ioc {
          type string;
          description
            "This field represents the Indicators of Compromise (IOC),
```

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

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

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

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

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

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

Figure 18: YANG for Consumer-Facing Interface

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

7.2. Scenario 1: Block SNS Access during Business Hours

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

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

Figure 21: An XML Example for Time-based Firewall

Time-based-condition Firewall

1. The policy name is "security_policy_for_blocking_sns".

2. The rule name is "block_access_to_sns_during_office_hours".

3. The Source is "employees".

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

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

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

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

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

Figure 22: An XML Example for VoIP Security Service

Custom-condition Firewall

1. The policy name is "security_policy_for_blocking_malicious_voip_packets".

2. The rule name is "Block_malicious_voip_and_vocn_packets".

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

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

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

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

The third example scenario is to "Mitigate flood attacks on a company web server" using a DDoS-attack mitigation policy. Here, the time information is not set because the service provided by the network should be maintained at all times. If the packets sent by any sources that target "webservers" are more than the set threshold, then the admin can set the percentage of the packets to be dropped to safely maintain the service. Once the rule is set and delivered and enforced to the NSFs by the security controller, the NSFs will monitor the incoming packet amounts to act according to the rule set. The XML instance is described below:

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

Figure 23: An XML Example for DDoS-attack Mitigation

DDoS-condition Firewall

1. The policy name is "security_policy_for_ddos_attacks".

2. The rule name is "1000_packets_per_second".

3. The destination is webservers.

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

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

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

8. IANA Considerations

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

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

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

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

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

9. Security Considerations

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

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

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

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

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

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

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

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

* container endpoint-groups: This node holds a list of endpoint data that may be considered private to the users. Disclosure of this information may expose sensitive details which can be used to define the identity and geographical location of a user. Malicious actors can leverage this information to threaten the user with cyber threat, e.g., voice phishing, or physical threat.

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

10. References

10.1. Normative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

10.2. Informative References

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

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

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

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

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

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

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

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

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

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

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

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

Appendix A. Acknowledgments

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

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

Appendix B. Contributors

The following are co-authors of this document:

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

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

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

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

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

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

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

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

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

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

Appendix C. Changes from draft-ietf-i2nsf-consumer-facing-interface-dm-30

The following changes are made from draft-ietf-i2nsf-consumer-facing-interface-dm-30:

* The usage of "hostnames" is removed from Section 4.4 as the URL cannot be given as a hostname. This update follows the comment of Lars Eggert.

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Chaehong Chung
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: darkhong@skku.edu

Tae-Jin Ahn
Korea Telecom
70 Yuseong-Ro, Yuseong-Gu
Daejeon
305-811
Republic of Korea
Phone: +82 42 870 8409
Email: taejin.ahn@kt.com

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
United States of America
Email: rkkumar@juniper.net

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America
Phone: +1-734-604-0332
Email: shares@ndzh.com

I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 December 2022

J. Kim, Ed.
J. Jeong, Ed.
Sungkyunkwan University
J. Park
ETRI
S. Hares
Q. Lin
Huawei

1 June 2022

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.

Table of Contents

1. Introduction

This document defines a YANG [RFC6020][RFC7950] data model for security policy rule configuration of Network Security Functions (NSF). The YANG data model in this document is based on the data model described in [I-D.ietf-i2nsf-capability-data-model] for the NSF-Facing Interface in the Interface to Network Security Functions (I2NSF) architecture [RFC8329]. The YANG data model in this document focuses on security policy configuration for the NSFs discussed in

[I-D.ietf-i2nsf-capability-data-model], i.e., generic NSF (operate on packet header for layer 2, layer3, and layer 4) and advanced NSF (Intrusion Prevention System, URL-Filtering, anti-DDoS, Antivirus, and VoIP/VoCN Filter). Note: VoIP is an abbreviation for Voice over Internet Protocol and VoCN is an abbreviation for Voice over Cellular Network, such as Voice over LTE or 5G.

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

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

* A security policy rule of a network security function.

* An event clause of a generic network security function.

* A condition clause of a generic network security function.

* An action clause of a generic network security function.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

This document uses the terminology described in [RFC8329].

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

3. YANG Tree Diagram

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

3.1. General I2NSF Security Policy Rule

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

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

Figure 1: YANG Tree Diagram for Network Security Policy

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

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

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

strategy can be extended according to specific vendor action features. The resolution strategy is described in detail in [I-D.ietf-i2nsf-capability-data-model].

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

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

3.2. Event Clause

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

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

Figure 2: YANG Tree Diagram for an Event Clause

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

3.3. Condition Clause

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

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

```
          |  |  |  |        +--:(source-ipv4-range)
          |  |  |  |           +--rw source-ipv4-range* [start end]
          |  |  |  |              +--rw start    inet:ipv4-address-no-zone
          |  |  |  |              +--rw end      inet:ipv4-address-no-zone
          |  |  |     +--:(ipv6)
          |  |  |        +--rw ipv6
          |  |  |           +--rw description?                      string
          |  |  |           +--rw dscp?                             inet:dscp
          |  |  |           +--rw ecn?                              uint8
          |  |  |           +--rw length?                           uint16
          |  |  |           +--rw ttl?                              uint8
          |  |  |           +--rw protocol?                         uint8
          |  |  |           +--rw (destination-network)?
          |  |  |           |  +--:(destination-ipv6-network)
          |  |  |           |  |  +--rw destination-ipv6-network?
                                                      inet:ipv6-prefix
          |  |  |           |  +--:(destination-ipv6-range)
          |  |  |           |     +--rw destination-ipv6-range* [start end]
          |  |  |           |        +--rw start    inet:ipv6-address-no-zone
          |  |  |           |        +--rw end      inet:ipv6-address-no-zone
          |  |  |           +--rw (source-network)?
          |  |  |           |  +--:(source-ipv6-network)
          |  |  |           |  |  +--rw source-ipv6-network? inet:ipv6-prefix
          |  |  |           |  +--:(source-ipv6-range)
          |  |  |           |     +--rw source-ipv6-range* [start end]
          |  |  |           |        +--rw start    inet:ipv6-address-no-zone
          |  |  |           |        +--rw end      inet:ipv6-address-no-zone
          |  |  |           +--rw flow-label?      inet:ipv6-flow-label
          |  |  +--rw (layer-4)?
          |  |  |  +--:(tcp)
          |  |  |  |  +--rw tcp
          |  |  |  |     +--rw description?               string
          |  |  |  |     +--rw source-port-number
          |  |  |  |     |  +--rw (source-port)?
          |  |  |  |     |     +--:(range-or-operator)
          |  |  |  |     |     |  +--rw (port-range-or-operator)?
          |  |  |  |     |     |     +--:(range)
          |  |  |  |     |     |     |  +--rw lower-port  inet:port-number
          |  |  |  |     |     |     |  +--rw upper-port  inet:port-number
          |  |  |  |     |     |     +--:(operator)
          |  |  |  |     |     |        +--rw operator?     operator
          |  |  |  |     |     |        +--rw port        inet:port-number
          |  |  |  |     |     +--:(port-list)
          |  |  |  |     |        +--rw port-numbers* [start end]
          |  |  |  |     |           +--rw start    inet:port-number
          |  |  |  |     |           +--rw end      inet:port-number
          |  |  |  |     +--rw destination-port-number
          |  |  |  |     |  +--rw (destination-port)?
```

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

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

```
          |  |  |  |     |     |        +--rw operator?     operator
          |  |  |  |     |     |        +--rw port        inet:port-number
          |  |  |  |     |     +--:(port-list)
          |  |  |  |     |        +--rw port-numbers* [start end]
          |  |  |  |     |           +--rw start    inet:port-number
          |  |  |  |     |           +--rw end      inet:port-number
          |  |  |  |     +--rw destination-port-number
          |  |  |  |     |  +--rw (destination-port)?
          |  |  |  |     |     +--:(range-or-operator)
          |  |  |  |     |     |  +--rw (port-range-or-operator)?
          |  |  |  |     |     |     +--:(range)
          |  |  |  |     |     |     |  +--rw lower-port  inet:port-number
          |  |  |  |     |     |     |  +--rw upper-port  inet:port-number
          |  |  |  |     |     |     +--:(operator)
          |  |  |  |     |     |        +--rw operator?     operator
          |  |  |  |     |     |        +--rw port        inet:port-number
          |  |  |  |     |     +--:(port-list)
          |  |  |  |     |        +--rw port-numbers* [start end]
          |  |  |  |     |           +--rw start    inet:port-number
          |  |  |  |     |           +--rw end      inet:port-number
          |  |  |  |     +--rw service-code*              uint32
          |  |  |  |     +--rw type*                      uint8
          |  |  |  |     +--rw data-offset?               uint8
          |  |  |  +--:(icmp)
          |  |  |     +--rw icmp
          |  |  |        +--rw description?      string
          |  |  |        +--rw version?          enumeration
          |  |  |        +--rw type?             uint8
          |  |  |        +--rw code?             uint8
          |  |  |        +--rw rest-of-header?   binary
          |  |  +--rw url-category
          |  |  |  +--rw description?    string
          |  |  |  +--rw pre-defined*    string
          |  |  |  +--rw user-defined*   string
          |  |  +--rw voice
          |  |  |  +--rw description?            string
          |  |  |  +--rw source-voice-id*        string
          |  |  |  +--rw destination-voice-id*   string
          |  |  |  +--rw user-agent*             string
          |  |  +--rw ddos
          |  |  |  +--rw description?         string
          |  |  |  +--rw alert-packet-rate?   uint32
          |  |  |  +--rw alert-flow-rate?     uint32
          |  |  |  +--rw alert-byte-rate?     uint32
          |  |  +--rw anti-virus
          |  |  |  +--rw profile*           string
          |  |  |  +--rw exception-files*   string
          |  |  +--rw payload
```

```
        |  |  |  +--rw description?   string
        |  |  |  +--rw content*       binary
        |  |  +--rw context
        |  |     +--rw description?           string
        |  |     +--rw time
        |  |     |  +--rw start-date-time?   yang:date-and-time
        |  |     |  +--rw end-date-time?     yang:date-and-time
        |  |     |  +--rw period
        |  |     |  |  +--rw start-time?   time
        |  |     |  |  +--rw end-time?     time
        |  |     |  |  +--rw day*          day
        |  |     |  |  +--rw date*         int8
        |  |     |  |  +--rw month*        string
        |  |     |  +--rw frequency?         enumeration
        |  |     +--rw application
        |  |     |  +--rw description?   string
        |  |     |  +--rw protocol*      identityref
        |  |     +--rw device-type
        |  |     |  +--rw description?   string
        |  |     |  +--rw device*        identityref
        |  |     +--rw users
        |  |     |  +--rw description?   string
        |  |     |  +--rw user* [id]
        |  |     |  |  +--rw id      uint32
        |  |     |  |  +--rw name?   string
        |  |     |  +--rw group* [id]
        |  |     |     +--rw id      uint32
        |  |     |     +--rw name?   string
        |  |     +--rw geographic-location
        |  |        +--rw description?   string
        |  |        +--rw source*        string
        |  |        +--rw destination*   string
        |  +--rw action
        |     ...
        +--rw rule-group
           ...
```

Figure 3: YANG Tree Diagram for a Condition Clause

A condition clause is defined as a set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to determine whether the set of actions in that (imperative) I2NSF policy rule can be executed or not. A condition clause works with 'AND' logic, where all fields set in the condition MUST match the packet or flow for the condition to be evaluated as 'TRUE'. A condition clause is classified as a

condition of generic network security functions, advanced network security functions, or context. A condition clause of generic network security functions is defined as IPv4 condition, IPv6 condition, TCP condition, UDP condition, SCTP condition, DCCP condition, or ICMP (ICMPv4 and ICMPv6) condition.

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

The data model provides transport layer condition for TCP, UDP, SCTP, and DCCP. With ICMPv4 and ICMPv6 are included as a choice for layer 4 as the header fields in ICMP are above the network layer. Note that QUIC protocol [RFC9000] is excluded in the data model as it is not considered in the initial I2NSF documents [RFC8329]. The QUIC traffic should not be treated as UDP traffic and will be considered in the future I2NSF documents.

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

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

3.4. Action Clause

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

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

Figure 4: YANG Tree Diagram for an Action Clause

An action is used to control and monitor aspects of flow-based NSFs when the policy rule event and condition clauses are satisfied. NSFs provide security services by executing various actions. The action clause is defined as ingress action, egress action, or log action for packet action, flow action, and advanced action for additional inspection. The packet action is an action for an individual packet such as an IP datagram as a stateless process that uses the packet's header and payload. The flow action is an action of a traffic flow such as the packets of a TCP session (e.g., an HTTP/HTTPS session) as a stateful process that uses the traffic flow information such as 5-tuple information, packet counts, and byte counts. The advanced action is an action for an advanced security service (e.g., url filter, DDoS-attack mitigator, and VoIP/VoCN filter) for either a packet or a traffic flow according to the intention of such an advanced security service. The action clause can be extended according to specific vendor action features. The action clause is described in detail in [I-D.ietf-i2nsf-capability-data-model].

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

4. YANG Data Model of NSF-Facing Interface

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

This data model is designed to support the I2NSF framework that can be extended according to the security needs. In other words, the model design is independent of the content and meaning of specific policies as well as the implementation approach.

With the YANG data model of I2NSF NSF-Facing Interface, this document suggests use cases for security policy rules such as time-based firewall, web filter, VoIP/VoCN security service, and DDoS-attack mitigation in Section 5.

4.1. YANG Module of NSF-Facing Interface

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

```
<CODE BEGINS> file "ietf-i2nsf-nsf-facing-interface@2022-06-01.yang"
module ietf-i2nsf-nsf-facing-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface";
  prefix
    i2nsfnfi;

  import ietf-inet-types {
    prefix inet;
```

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

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

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

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

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

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

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
       'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
       'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this
       document are to be interpreted as described in BCP 14
       (RFC 2119) (RFC 8174) when, and only when, they appear
       in all capitals, as shown here.

       Copyright (c) 2022 IETF Trust and the persons identified as
       authors of the code. All rights reserved.

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

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

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

     /*
      * Identities
      */

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

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

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

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

     identity system-event {
       base event;
```

```
      description
        "Base Identity for system events. System event (also called
         alert) is defined as a warning about any changes of
         configuration, any access violation, the information of
         sessions and traffic flows.";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
         Monitoring Interface YANG Data Model - System event";
    }

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

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

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

    identity memory-alarm {
      base system-alarm;
      description
```

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

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

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

    identity hardware-alarm {
      base system-alarm;
      description
        "A hardware alarm is emitted when a hardware failure (e.g.,
         CPU, memory, disk, or interface) is detected. A hardware
         failure is a malfunction within the electronic circuits or
         electromechanical components of the hardware that makes it
         unusable.";
      reference
        "draft-ietf-i2nsf-nsf-monitoring-data-model-20: I2NSF NSF
         Monitoring Interface YANG Data Model - System alarm for
         hardware";
    }

    identity interface-alarm {
```

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

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


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

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

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

```
     }

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

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

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

     identity ot {
       base device-type;
       description
         "Identity for Operational Technology (OT) devices (also
          known as industrial control systems) that interact
          with the physical environment and detect or cause direct
          change through the monitoring and control of devices,
          processes, and events such as programmable logic
          controllers (PLCs), digital oscilloscopes, building
          management systems (BMS), and fire control systems";
     }

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

     identity advanced-nsf {
       description
         "Base identity for advanced Network Security Function (NSF)
          capability.  This can be used for advanced NSFs such as
          Anti-DDoS Attack, IPS, URL-Filtering, Antivirus,
```

```
          and VoIP/VoCN Filter.";
      reference
        "draft-ietf-i2nsf-capability-data-model-32:
         I2NSF Capability YANG Data Model";
    }

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

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

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

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

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

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

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

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

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

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

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

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

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

     identity reject {
       base ingress-action;
       base egress-action;
       base default-action;
       description
         "The reject action denies a packet to go through the NSF
          entering or exiting the internal network and sends a response
          back to the source. The response depends on the packet and
          implementation. For example, a TCP packet is rejected with
          TCP RST response or a UDP packet may be rejected with an
```

```
          ICMPv4 response message with Type 3 Code 3 or ICMPv6 response
          message Type 1 Code 4 (i.e., Destination Unreachable:
          Destination port unreachable).";
     }

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

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

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

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

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

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

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

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

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

     identity resolution-strategy {
```

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

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

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

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

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

    identity application-protocol {
      description
        "Base identity for Application protocol. Note that a subset of
```

```
          application protocols (e.g., HTTP, HTTPS, FTP, POP3, and
          IMAP) are handled in this YANG module, rather than all
          the existing application protocols.";
     }

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

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

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

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

     identity ftp {
       base application-protocol;
       description
         "The identity for File Transfer Protocol.";
       reference
```

```
        "RFC 959: File Transfer Protocol (FTP)";
    }

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

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

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

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

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

    identity imap {
      base application-protocol;
      description
        "The identity for Internet Message Access Protocol (IMAP).";
```

```
      reference
        "RFC 9051: Internet Message Access Protocol (IMAP) - Version
         4rev2";
    }

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

    /*
     * Typedefs
     */

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

    typedef day {
      type enumeration {
        enum monday {
          description
            "This represents Monday.";
        }
        enum tuesday {
          description
            "This represents Tuesday.";
        }
        enum wednesday {
          description
            "This represents Wednesday";
        }
        enum thursday {
          description
            "This represents Thursday.";
        }
        enum friday {
```

```
          description
            "This represents Friday.";
        }
        enum saturday {
          description
            "This represents Saturday.";
        }
        enum sunday {
          description
            "This represents Sunday.";
        }
      }
      description
        "The type for representing the day of the week.";
    }

    /*
     * Groupings
     */

    grouping port-range {
      leaf start {
        type inet:port-number;
        description
          "A start port number for a range match.";
      }
      leaf end {
        type inet:port-number;
        must '. >= ../start' {
          error-message
            "An end port number MUST be equal to or greater than a
             start port number.";
        }
        description
          "An end port number for a range match.";
      }
      description
        "A range match for port numbers. If only one value is needed,
         then set both start and end to the same value.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification - Port Number
         RFC 768: User Datagram Protocol - Port Number
         draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
         Protocol - Port Number
         RFC 4340: Datagram Congestion Control Protocol (DCCP)
         - Port Number";
    }
```

```
     grouping ipv4-range {
       description
         "A range match for IPv4 addresses.  If only one value is
          needed, then set both start and end to the same value.
          The end IPv4 address MUST be equal to or greater than the
          start IPv4 address.";
       leaf start {
         type inet:ipv4-address-no-zone;
         description
           "A start IPv4 address for a range match.";
       }
       leaf end {
         type inet:ipv4-address-no-zone;
         description
           "An end IPv4 address for a range match.";
       }
       reference
         "RFC 791: Internet Protocol - IPv4 address";
     }

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

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

     /*
      * Data nodes
      */

     list i2nsf-security-policy {
```

```
      key "name";

      description
        "Container for security policy
         including a set of security rules according to certain logic,
         i.e., their similarity or mutual relations, etc. The network
         security policy can be applied to both the unidirectional
         and bidirectional traffic across the NSF.
         The I2NSF security policies use the Event-Condition-Action
         (ECA) policy model ";

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

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

      leaf language {
        type string {
          pattern '((([A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3})'
                + '{0,2})?)|[A-Za-z]{4}|[A-Za-z]{5,8})(-[A-Za-z]{4})?'
                + '(-([A-Za-z]{2}|[0-9]{3}))?(-([A-Za-z0-9]{5,8}'
                + '|([0-9][A-Za-z0-9]{3})))*(-[0-9A-WYZa-wyz]'
                + '(-([A-Za-z0-9]{2,8}))+)*(-[Xx](-([A-Za-z0-9]'
                + '{1,8}))+)?|[Xx](-([A-Za-z0-9]{1,8}))+|'
                + '(([Ee][Nn]-[Gg][Bb]-[Oo][Ee][Dd]|[Ii]-'
                + '[Aa][Mm][Ii]|[Ii]-[Bb][Nn][Nn]|[Ii]-'
                + '[Dd][Ee][Ff][Aa][Uu][Ll][Tt]|[Ii]-'
                + '[Ee][Nn][Oo][Cc][Hh][Ii][Aa][Nn]'
                + '|[Ii]-[Hh][Aa][Kk]|'
                + '[Ii]-[Kk][Ll][Ii][Nn][Gg][Oo][Nn]|'
                + '[Ii]-[Ll][Uu][Xx]|[Ii]-[Mm][Ii][Nn][Gg][Oo]|'
                + '[Ii]-[Nn][Aa][Vv][Aa][Jj][Oo]|[Ii]-[Pp][Ww][Nn]|'
                + '[Ii]-[Tt][Aa][Oo]|[Ii]-[Tt][Aa][Yy]|'
                + '[Ii]-[Tt][Ss][Uu]|[Ss][Gg][Nn]-[Bb][Ee]-[Ff][Rr]|'
                + '[Ss][Gg][Nn]-[Bb][Ee]-[Nn][Ll]|[Ss][Gg][Nn]-'
                + '[Cc][Hh]-[Dd][Ee])|([Aa][Rr][Tt]-'
                + '[Ll][Oo][Jj][Bb][Aa][Nn]|[Cc][Ee][Ll]-'
                + '[Gg][Aa][Uu][Ll][Ii][Ss][Hh]|'
                + '[Nn][Oo]-[Bb][Oo][Kk]|[Nn][Oo]-'
```

```
                 + '[Nn][Yy][Nn]|[Zz][Hh]-[Gg][Uu][Oo][Yy][Uu]|'
                 + '[Zz][Hh]-[Hh][Aa][Kk][Kk][Aa]|[Zz][Hh]-'
                 + '[Mm][Ii][Nn]|[Zz][Hh]-[Mm][Ii][Nn]-'
                 + '[Nn][Aa][Nn]|[Zz][Hh]-[Xx][Ii][Aa][Nn][Gg])))';
          }
          default "en-US";
          description
            "The value in this field indicates the language tag
             used for all of the 'leaf description' described in the
             'i2nsf-security-policy'. This field is mandatory only when
             one or more of the 'leaf description' is used.

             The attribute is encoded following the rules in Section 2.1
             in RFC 5646. The default language tag is 'en-US'";
          reference
            "RFC 5646: Tags for Identifying Languages";
        }

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

        leaf resolution-strategy {
          type identityref {
            base resolution-strategy;
          }
          default fmr;
          description
            "The resolution strategies that can be used to
             specify how to resolve conflicts that occur between
             actions of the same or different policy rules that
             are matched and contained in this particular NSF";

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

        leaf default-action {
          type identityref {
            base default-action;
          }
```

```
        default mirror;
        description
          "This default action can be used to specify a predefined
           action when no other alternative action was matched
           by the currently executing I2NSF Policy Rule. An analogy
           is the use of a default statement in a C switch statement.";
        reference
          "draft-ietf-i2nsf-capability-data-model-32:
           I2NSF Capability YANG Data Model - Default Action";
      }

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

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

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

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

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

        container long-connection {
```

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

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

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

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

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

          leaf description {
```

```
            type string;
            description
              "Description for an event clause";
          }

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

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

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

          leaf description {
            type string;
            description
```

```
            "Description for a condition clause.";
          }

          list layer-2 {
            key "destination-mac-address source-mac-address ethertype";
            description
              "The purpose of this container is to represent layer 2
               packet header information to determine the set of policy
               actions in this ECA policy rule should be executed or
               not.";
            reference
              "IEEE 802.3: IEEE Standard for Ethernet";

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

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

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

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

                uses packet-fields:acl-ip-header-fields;
                uses packet-fields:acl-ipv4-header-fields {
                  augment destination-network {
                    case destination-ipv4-range {
                      list destination-ipv4-range {
                        key "start end";
                        uses ipv4-range;
                        description
                          "The list of IPv4 addresses specified with
```

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

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

               leaf description {
                 type string;
```

```
                  description
                    "This is description for IPv6 condition.";
                }

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

```
              "Choice of either IPv4 or IPv6 as layer-3 protocol";
          }

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


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

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

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

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

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

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

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

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

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

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

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

                leaf description {
```

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

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

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

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

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

                leaf chunk-length {
                  type uint16 {
                    range "4..max";
                  }
                  description
                    "The security policy rule according to the length
                     of  the chunk in sctp. This value represents the
                     size of the chunk in bytes, including the Chunk
                     Type, Chunk Flags, Chunk Length, and Chunk Value
                     fields.";
                  reference
```

```
                    "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control
                     Transmission Protocol - Chunk Length";
                }
              }
            }

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

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

```
                  reference
                    "RFC 4340: Datagram Congestion Control Protocol
                               (DCCP) - Port number";
                }

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

                leaf-list service-code {
                  type uint32;
                  description
                    "The security policy rule according to
                     dccp service code.";
                  reference
                    "RFC 4340: Datagram Congestion Control Protocol
                               (DCCP) - Service Codes
```

```
                     RFC 5595: The Datagram Congestion Control Protocol
                               (DCCP) Service Codes
                     RFC 6335: Internet Assigned Numbers Authority
                               (IANA) Procedures for the Management of
                               the Service Name and Transport Protocol
                               Port Number Registry - Service Code";
                }

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

                leaf data-offset {
                  type uint8;
                  description
                    "The security policy rule according to the offset
                     from
                     the start of the packet's DCCP header to the start
                     of its application data area, in 32-bit word.";
                  reference
                    "RFC 4340: Datagram Congestion Control Protocol
                               (DCCP) - Data Offset";
                }
              }
            }
            case icmp {
              container icmp {
                description
                  "The purpose of this container is to represent
                   ICMPv4 and ICMPv6 packet header information to
                   determine if the set of policy actions in this ECA
                   policy rule should be executed or not.";
                reference
                  "RFC  792: Internet Control Message Protocol
                   RFC 8335: PROBE: A Utility for Probing Interfaces";

                leaf description {
                  type string;
```

```
                  description
                   "This is description for icmp condition.";
                }

                leaf version {
                  type enumeration {
                    enum icmpv4 {
                      value "1";
                      description
                        "The ICMPv4 Protocol as defined in RFC 792";
                    }
                    enum icmpv6 {
                      value "2";
                      description
                        "The ICMPv6 Protocol as defined in RFC 4443";
                    }
                  }
                  description
                    "The ICMP version to be matched. This value
                     affected the type and code values.";
                  reference
                    "RFC  792: Internet Control Message Protocol
                     RFC 4443: Internet Control Message Protocol
                               (ICMPv6) for the Internet Protocol
                               Version 6 (IPv6) Specification";
                }

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

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

            leaf-list pre-defined {
              type string;
```

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

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

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

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

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

```
            }

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

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

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

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

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

            leaf alert-byte-rate {
              type uint32;
              units "Bps";
              description
                "The alert rate of flood detection for
```

```
                 bytes per second (Bps) of an IP address.
                 If the bytes per second of an IP address
                 exceeds the alert rate threshold, an alert
                 will be generated.";
             }
           }

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

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

             leaf-list exception-files {
               type string;
               description
                 "The type or name of the files to be excluded by the
                  antivirus. This can be used to keep the known
                  harmless files. Absolute paths are filenames/paths
                  to be excluded and relative ones are interpreted as
                  globs.";
               reference
                 "GLOB: Linux Programmer's Manual - GLOB";
             }
           }

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

```
            }
          }

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

            container time {
              description
                "Time to determine when the policy should be applied";
              leaf start-date-time {
                type yang:date-and-time;
                description
                  "This is the start date and time for a security
                   policy rule.";
              }

              leaf end-date-time {
                type yang:date-and-time;
                description
                  "This is the end date and time for a policy rule.
                   The policy rule will stop working after the
                   specified end-date-time.";
              }

              container period {
                when
                  "../frequency!='only-once'";
                description
                  "This represents the repetition time.  In the case
                   where the frequency is weekly, the days can be
                   set.";
                leaf start-time {
                  type time;
                  description
                    "This is a period's start time for an event.";
                }
                leaf end-time {
                  type time;
                  description
                    "This is a period's end time for an event.";
                }
                leaf-list day {
```

```
                when
                  "../../frequency='weekly'";
                type day;
                min-elements 1;
                description
                  "This represents the repeated day of every week
                   (e.g., Monday and Tuesday).  More than one day
                   can be specified.";
              }
              leaf-list date {
                when
                  "../../frequency='monthly'";
                type int8 {
                  range "1..31";
                }
                min-elements 1;
                description
                  "This represents the repeated date of every month.
                   More than one date can be specified.";
              }
              leaf-list month {
                when
                  "../../frequency='yearly'";
                type string{
                  pattern '\d{2}-\d{2}';
                }
                min-elements 1;
                description
                  "This represents the repeated date and month of
                   every year.  More than one can be specified.
                   A pattern used here is Month and Date (MM-DD).";
              }
            }

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

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

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

          container device-type {
            description
```

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

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

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

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

            container geographic-location {
              description
                "The location which network traffic flow is associated
                 with. The region can be the geographic location such
                 as country, province, and city, as well as the logical
                 network location such as IP address, network section,
                 and network domain.";
              reference
                "RFC 9179: A YANG Grouping for Geographic Locations";

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

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

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

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

          container action {
            description
              "An action is used to control and monitor aspects of
               flow-based NSFs when the event and condition clauses
               are satisfied. NSFs provide security functions by
               executing various Actions. Examples of I2NSF Actions
               include providing intrusion detection and/or protection,
               web and flow filtering, and deep packet inspection
               for packets and flows.";
            reference
              "RFC 8329: Framework for Interface to Network Security
               Functions - I2NSF Flow Security Policy Structure
               draft-ietf-i2nsf-capability-data-model-32:
               I2NSF Capability YANG Data Model - Design Principles and
               ECA Policy Model Overview";

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

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

```
                ECA Policy Model Overview";

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

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

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

          }

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

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

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

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

          container advanced-action {
            description
              "If the packet needs to be additionally inspected,
               the packet is passed to advanced network
               security functions according to the profile.
               The profile means the types of NSFs where the packet
               will be forwarded in order to additionally
               inspect the packet.
               The advanced action activates Service Function
               Chaining (SFC) for further inspection of a packet.";
            reference
              "draft-ietf-i2nsf-capability-data-model-32:
               I2NSF Capability YANG Data Model - YANG Tree
               Diagram";

            leaf-list content-security-control {
              type identityref {
                base content-security-control;
              }
              description
                "Content-security-control is the NSFs that
                 inspect the payload of the packet.
                 The profile for the types of NSFs for mitigation is
                 divided into content security control and
                 attack-mitigation-control.
                 Content security control: ips, url filtering,
                 antivirus, and voip-vocn-filter. This can be
                 extended according to the provided NSFs.";
              reference
```

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

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

        list groups {
          key "group-name";
          description
            "This is a group for rules";

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

          leaf-list rule-name {
            type leafref {
              path
                "../../../rules/name";
            }
            description
```

```
                "The names of the rules to be grouped.";
            }

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

            leaf description {
              type string;
              description
                "This is a description for rule-group";
            }
          }
        }
      }
    }
    <CODE ENDS>
```

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

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

This section shows XML configuration examples of low-level security policy rules that are delivered from the Security Controller to NSFs over the NSF-Facing Interface. For security requirements, we assume that the NSFs (i.e., General firewall, Time-based firewall, URL filter, VoIP/VoCN filter, and HTTP and HTTPS flood mitigation) described in Appendix A of [I-D.ietf-i2nsf-capability-data-model] are registered with the I2NSF framework. With the registered NSFs, we show configuration examples for security policy rules of network security functions according to the following three security requirements: (i) Block Social Networking Service (SNS) access during business hours, (ii) Block malicious VoIP/VoCN packets coming to the company, and (iii) Mitigate HTTP and HTTPS flood attacks on company web server.

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

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

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>sns_access</name>
 <rules>
  <name>block_sns_access_during_operation_time_for_ipv4</name>
  <condition>
   <ipv4>
    <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
   </ipv4>
   <context>
    <time>
      <start-date-time>2021-03-11T09:00:00.00Z</start-date-time>
      <end-date-time>2021-12-31T18:00:00.00Z</end-date-time>
      <period>
        <start-time>09:00:00Z</start-time>
        <end-time>18:00:00Z</end-time>
        <day>monday</day>
        <day>tuesday</day>
        <day>wednesday</day>
        <day>thursday</day>
        <day>friday</day>
      </period>
      <frequency>weekly</frequency>
    </time>
   </context>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

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

```
<i2nsf-security-policy
xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-nsf-facing-interface">
 <name>sns_access</name>
 <rules>
  <name>block_sns_access_during_operation_time_for_ipv6</name>
  <condition>
   <ipv6>
    <source-ipv6-network>2001:db8:1::/60</source-ipv6-network>
   </ipv6>
   <context>
    <time>
      <start-date-time>2021-03-11T09:00:00.00Z</start-date-time>
      <end-date-time>2021-12-31T18:00:00.00Z</end-date-time>
      <period>
        <start-time>09:00:00Z</start-time>
        <end-time>18:00:00Z</end-time>
        <day>monday</day>
        <day>tuesday</day>
        <day>wednesday</day>
        <day>thursday</day>
        <day>friday</day>
      </period>
      <frequency>weekly</frequency>
    </time>
   </context>
  </condition>
  <action>
   <advanced-action>
    <content-security-control>
      url-filtering
    </content-security-control>
   </advanced-action>
  </action>
 </rules>
</i2nsf-security-policy>
```

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

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

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

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

Time-based Firewall is as follows:

1. The name of the security policy is sns_access.

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

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

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

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

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

Web Filter is as follows:

1. The name of the security policy is sns_access.

2. The name of the rule is block_SNS_1_and_SNS_2.

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

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

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

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

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

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

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

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

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

General Firewall is as follows:

1. The name of the security policy is voip_vocn_inspection.

2. The name of the rule is block_malicious_voice_id.

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

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

5. If the incoming packets match the rules above, the general firewall sends the packets to VoIP/VoCN filter for additional inspection because the general firewall can not inspect contents of the VoIP/VoCN packets.

VoIP/VoCN Filter is as follows:

1. The name of the security policy is malicious_voice_id.

2. The name of the rule is block_malicious_voice_id.

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

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

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

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

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

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

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

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

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

General Firewall is as follows:

1. The name of the security policy is flood_attack_mitigation.

2. The name of the rule is mitigate_http_and_https_flood_attack.

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

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

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

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

1. The name of the security policy is flood_attack_mitigation.

2. The name of the rule is mitigate_http_and_https_flood_attack.

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

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

6. IANA Considerations

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

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

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

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

7. Security Considerations

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

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

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable

in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

* ietf-i2nsf-nsf-facing-interface: Writing to almost any element of this YANG module would directly impact on the configuration of NSFs, e.g., completely turning off security monitoring and mitigation capabilities; altering the scope of this monitoring and mitigation; creating an overwhelming logging volume to overwhelm downstream analytics or storage capacity; creating logging patterns which are confusing; or rendering useless trained statistics or artificial intelligence models.

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

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

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

8. References

8.1. Normative References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

8.2. Informative References

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

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

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

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

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

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

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

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

Appendix A. Acknowledgments

This document is a product by the I2NSF Working Group (WG) including WG Chairs (i.e., Linda Dunbar and Yoav Nir) and Diego Lopez. This document took advantage of the review and comments from the following people: Roman Danyliw, Acee Lindem, Dan Romascanu (GenART), Yoshifumi Nishida (TSVART), Kyle Rose (SecDir), Joe Clarke (OpsDir), and Tom Petch. The authors sincerely appreciate their sincere efforts and kind help.

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

Appendix B. Contributors

The following are co-authors of this document:

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

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

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

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

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

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

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

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

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

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

Authors' Addresses

Jinyong Tim Kim (editor)
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 10 8273 0930
Email: timkim@skku.edu

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon
34129
Republic of Korea
Phone: +82 42 860 6514
Email: pjs@etri.re.kr

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America
Phone: +1-734-604-0332
Email: shares@ndzh.com

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen
Guangdong 518129,
China
Email: linqiushi@huawei.com

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: 3 December 2022

J. Jeong, Ed.
P. Lingga
Sungkyunkwan University
S. Hares
L. Xia
Huawei
H. Birkholz
Fraunhofer SIT
1 June 2022

I2NSF NSF Monitoring Interface YANG Data Model
draft-ietf-i2nsf-nsf-monitoring-data-model-20

Abstract

This document proposes an information model and the corresponding YANG data model of an interface for monitoring Network Security Functions (NSFs) in the Interface to Network Security Functions (I2NSF) framework. If the monitoring of NSFs is performed with the NSF monitoring interface in a standard way, it is possible to detect the indication of malicious activity, anomalous behavior, the potential sign of denial-of-service attacks, or system overload in a timely manner. This monitoring functionality is based on the monitoring information that is generated by NSFs. Thus, this document describes not only an information model for the NSF monitoring interface along with a YANG tree diagram, but also the corresponding YANG data model.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

According to [RFC8329], the interface provided by a Network Security Function (NSF) (e.g., Firewall, IPS, or Anti-DDoS function) to enable the collection of monitoring information is referred to as an I2NSF Monitoring Interface. This interface enables the sharing of vital data from the NSFs (e.g., events, records, and counters) to an NSF data collector (e.g., Security Controller) through a variety of mechanisms (e.g., queries and notifications). The monitoring of NSF plays an important role in an overall security framework, if it is done in a timely way. The monitoring information generated by an NSF can be a good, early indication of anomalous behavior or malicious activity, such as denial-of-service (DoS) attacks.

This document defines an information model of an NSF monitoring interface that provides visibility into an NSF for the NSF data collector (note that an NSF data collector is defined as an entity to collect NSF monitoring data from an NSF, such as Security Controller). It specifies the information and illustrates the methods that enable an NSF to provide the information required in order to be monitored in a scalable and efficient way via the NSF Monitoring Interface. The information model for the NSF monitoring interface presented in this document is complementary for the security policy provisioning functionality of the NSF-Facing Interface specified in [I-D.ietf-i2nsf-nsf-facing-interface-dm].

This document also defines a YANG [RFC7950] data model for the NSF monitoring interface, which is derived from the information model for the NSF monitoring interface.

Note that this document covers a subset of monitoring data for systems and NSFs, which are related to security.

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]. In addition, the following terms are defined in this document:

* I2NSF User: An entity that delivers a high-level security policy to the Security Controller and may request monitoring information via the NSF data collector.

* Monitoring Information: Relevant data that can be processed to know the status and performance of the network and the NSF. The monitoring information in an I2NSF environment consists of I2NSF Events, I2NSF Records, and I2NSF Counters (see Section 4.1 for the detailed definition). This information is to be delivered to the NSF data collector.

* Notification: Unsolicited transmission of monitoring information.

* NSF Data Collector: An entity that collects NSF monitoring information from NSFs, such as Security Controller.

* Subscription: An agreement initialized by the NSF data collector to receive monitoring information from an NSF. The method to subscribe follows the method by either NETCONF or RESTCONF, explained in [RFC5277] and [RFC8650], respectively.

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

3. Use Cases for NSF Monitoring Data

As mentioned earlier, monitoring plays a critical role in an overall security framework. The monitoring of the NSF provides very valuable information to an NSF data collector (e.g., Security Controller) in maintaining the provisioned security posture. Besides this, there are various other reasons to monitor the NSF as listed below:

* The I2NSF User that is the security administrator can configure a policy that is triggered on a specific event occurring in the NSF or the network [RFC8329] [I-D.ietf-i2nsf-consumer-facing-interface-dm]. If an NSF data collector (e.g., Security Controller) detects the specified event, it can configure additional security functions as defined by policies.

* The events triggered by an NSF as a result of security policy violation can be used by Security Information and Event Management (SIEM) to detect any suspicious activity in a larger correlation context.

* The information (i.e., events, records, and counters) from an NSF can be used to build advanced analytics, such as behavior and predictive models to improve security posture in large deployments.

* The NSF data collector can use events from the NSF for achieving high availability. It can take corrective actions such as restarting a failed NSF and horizontally scaling up the NSF.

* The information (i.e., events, records, and counters) from the NSF can aid in the root cause analysis of an operational issue, so it can improve debugging.

* The records from the NSF can be used to build historical data for operation and business reasons.

4. Classification of NSF Monitoring Data

In order to maintain a strong security posture, it is not only necessary to configure an NSF's security policies but also to continuously monitor the NSF by checking acquirable and observable data. This enables security administrators to assess the state of the networks in a timely fashion. It is not possible to block all the internal and external threats based on static security posture. A more practical approach is supported by enabling dynamic security measures, for which continuous visibility is required. This document defines a set of monitoring elements and their scopes that can be

acquired from an NSF and can be used as NSF monitoring data. In essence, this monitoring data can be leveraged to support constant visibility on multiple levels of granularity and can be consumed by the corresponding functions.

Three basic domains of monitoring data originating from a system entity [RFC4949], i.e., an NSF, are discussed in this document.

* Retention and Emission from NSFs

* Notifications for Events and Records

* Push and Pull for the retrieval of monitoring data from NSFs

Every system entity creates information about some context with defined I2NSF monitoring data, and so every system entity that provides such information can be an I2NSF component. This information is intended to be consumed by other I2NSF components, which deals with NSF monitoring data in an automated fashion.

4.1. Retention and Emission from NSFs

A system entity (e.g., NSF) first retains I2NSF monitoring data inside its own system before emitting the information to another I2NSF component (e.g., NSF Data Collector). The I2NSF monitoring information consist of I2NSF Events, I2NSF Records, and I2NSF Counters as follows:

I2NSF Event: I2NSF Event is defined as an important occurrence at a particular time, that is, a change in the system being managed or a change in the environment of the system being managed. An I2NSF Event requires immediate attention and should be notified as soon as possible. When used in the context of an (imperative) I2NSF Policy Rule, an I2NSF Event is used to determine whether the Condition clause of that Policy Rule can be evaluated or not. The Alarm Management Framework in [RFC3877] defines an event as something that happens which may be of interest. Examples of an event are a fault, a change in status, crossing a threshold, or an external input to the system. In the I2NSF domain, I2NSF events are created following the definition of an event in the Alarm Management Framework.

I2NSF Record: A record is defined as an item of information that is kept to be looked at and used in the future. Typically, records are the information, which is based on operational and informational data (i.e., various changes in system characteristics). They are generated by a system entity (e.g., NSF) at particular instants to be kept without any changes

afterward. A set of records has an ordering in time based on when they are generated. Unlike I2NSF Events, records do not require immediate attention but may be useful for visibility and retroactive cyber forensics. Records are typically stored in log-files or databases on a system entity or NSF. The examples of records include user activities, device performance, and network status. They are important for debugging, auditing, and security forensic of a system entity or the network having the system entity.

I2NSF Counter: An I2NSF Counter is defined as a specific representation of an information element whose value changes very frequently. Prominent examples are network interface counters for protocol data unit (PDU) amount, byte amount, drop counters, and error counters. Counters are useful in debugging and visibility into operational behavior of a system entity (e.g., NSF). When an NSF data collector asks for the value of a counter, a system entity MUST update the counter information and emit the latest information to the NSF data collector.

Retention is defined as the storing of monitoring data in NSFs. The retention of I2NSF monitoring information may be affected by the importance of the data. The importance of the data could be context-dependent, where it may not just be based on the type of data, but may also depend on where it is deployed, e.g., a test lab and testbed. The local policy and configuration will dictate the policies and procedures to review, archive, or purge the collected monitoring data.

Emission is defined as the delivery of monitoring data in NSFs to an NSF data collector. The I2NSF monitoring information retained on a system entity (e.g., NSF) may be delivered to a corresponding I2NSF User via an NSF data collector. The information consists of the aggregated records, typically in the form of log-files or databases. For the NSF Monitoring Interface to deliver the information to the NSF data collector, the NSF needs to accommodate standardized delivery protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. The NSF data collector can forward the information to the I2NSF User through standardized delivery protocols (e.g., RESTCONF and NETCONF). The interface for the delivery of Monitoring Data from the NSF data collector to the I2NSF User is out of the scope of this document.

4.2. Notifications for Events and Records

A specific task of an I2NSF User is to provide I2NSF Policy Rules. The rules of a policy are composed of three clauses: Event, Condition, and Action clauses. In consequence, an I2NSF Event is specified to trigger the evaluation of the Condition clause of the I2NSF Policy Rule. Such an I2NSF Event is defined as an important occurrence at a particular time in the system being managed, and/or in the environment of the system being managed whose concept aligns well with the generic definition of Event from [RFC3877].

Another role of the I2NSF Event is to trigger a notification for monitoring the status of an NSF. A notification is defined in [RFC3877] as an unsolicited transmission of management information. System alarm (called alarm) is defined as a warning related to service degradation in system hardware in Section 6.1. System event (called alert) is defined as a warning about any changes of configuration, any access violation, information about sessions and traffic flows in Section 6.2. Both an alarm and an alert are I2NSF Events that can be delivered as a notification. The model illustrated in this document introduces a complementary type of information that can be a conveyed notification.

In I2NSF monitoring, a notification is used to deliver either an event or a record via the I2NSF Monitoring Interface. The difference between the event and record is the timing by which the notifications are emitted. An event is emitted as soon as it happens in order to notify an NSF Data Collector of the problem that needs immediate attention. A record is not emitted immediately to the NSF Data Collector, and it can be emitted periodically to the NSF Data Collector.

It is important to note that an NSF Data Collector as a consumer (i.e., observer) of a notification assesses the importance of the notification rather than an NSF as a producer. The producer can include metadata in a notification that supports the observer in assessing its importance (e.g., severity).

4.3. Push and Pull for the retrieval of monitoring data from NSFs

An important aspect of monitoring information is the freshness of the information. From the perspective of security, it is important to notice changes in the current status of the network. The I2NSF Monitoring Interface provides the means of sending monitored information from the NSFs to an NSF data collector in a timely manner. Monitoring information can be acquired by a client (i.e., NSF data collector) from a server (i.e., NSF) using push [RFC5277] [RFC8641] or pull methods [RFC6241] [RFC8040].

The pull is a query-based method to obtain information from the NSF. In this method, the NSF will remain passive until the information is requested from the NSF data collector. Once a request is accepted (with proper authentication), the NSF MUST update the information before sending it to the NSF data collector.

The push is a report-based method to obtain information from the NSF. The report-based method ensures the information can be delivered immediately without any requests. This method is used by the NSF to actively provide information to the NSF data collector. To receive the information, the NSF data collector subscribes to the NSF for the information.

These acquisition methods are used for different types of monitoring information. The information that has a high level of urgency (i.e., I2NSF Event) should be provided with the push method, while information that has a lower level of urgency (i.e., I2NSF Record and I2NSF Counter) can be provided with either the pull method or push method.

5. Basic Information Model for Monitoring Data

As explained in the above section, there is a wealth of data available from NSFs that can be monitored. Firstly, there must be some general information with each monitoring message sent from an NSF that helps a consumer to identify metadata with that message, which are listed as below:

* message: The extra detailed description of NSF monitoring data to give an NSF data collector the context information as metadata.

* vendor-name: The vendor’s name of the NSF that generates the message.

* device-model: The model of the device, can be represented by the device model name or serial number. This field is used to identify the model of the device that provides the security service.

* software-version: The version of the software used to provide the security service.

* nsf-name: The name or IP address of the NSF generating the message. If the given nsf-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 a different NSF to identify the NSF that generates the message.

* timestamp: The time when the message was generated. For the notification operations (i.e., System Alarms, System Events, NSF Events, System Logs, and NSF Logs), this is represented by the eventTime of NETCONF event notification [RFC5277] For other operations (i.e., System Counter and NSF Counter), the timestamp MUST be provided separately. The time format used is following the rules in Section 5.6 of [RFC3339].

* language: describes the human language intended for the user, so that it allows a user to verify the language that is used in the notification (i.e., '../message', '/i2nsf-log/i2nsf-nsf-system-access-log/output', and '/i2nsf-log/i2nsf-system-user-activity-log/additional-info/cause'). The attribute is encoded following the rules in Section 2.1 of [RFC5646]. The default language tag is "en-US".

6. Extended Information Model for Monitoring Data

The extended information model is the specific monitoring data that covers the additional information associated with the detailed information of status and performance of the network and the NSF over the basic information model. The extended information combined with the basic information creates the monitoring information (i.e., I2NSF Event, Record, and Counter).

The extended monitoring information has settable characteristics for data collection as follows:

* Acquisition method: The method to obtain the message. It can be a "query" or a "subscription". A "query" is a request-based method to acquire the solicited information. A "subscription" is a report-based method that pushes information to the subscriber.

* Emission type: The cause type for the message to be emitted. This attribute is used only when the acquisition method is a "subscription" method. The emission type can be either "on-change" or "periodic". An "on-change" message is emitted when an important event happens in the NSF. A "periodic" message is emitted at a certain time interval. The time to periodically emit the message is configurable.

* Dampening type: The type of message dampening to stop the rapid transmission of messages. The dampening types are "on-repetition" and "no-dampening". The "on-repetition" type limits the transmitted "on-change" message to one message at a certain interval (e.g., 100 centiseconds). This interval is defined as dampening-period in [RFC8641]. The dampening-period is configurable in the unit of centiseconds. The "no-dampening" type

does not limit the transmission for the messages of the same type. In short, "on-repetition" means that the dampening is active and "no-dampening" is inactive. Activating the dampening for an "on-change" type of message is RECOMMENDED to reduce the number of messages generated.

Note that the characteristic information is not mandatory to be included in a monitoring message. The information is expected to be stored and may or may not be useful in some ways in the future. In any case, the inclusion of the characteristic information is up to the implementation.

6.1. System Alarms

System alarms have the following characteristics:

* acquisition-method: subscription

* emission-type: on-change

* dampening-type: on-repetition or no-dampening

6.1.1. Memory Alarm

The memory is the hardware to store information temporarily or for a short period, i.e., Random Access Memory (RAM). The memory-alarm is emitted when the memory usage exceeds the threshold. The following information should be included in a Memory Alarm:

* event-name: memory-alarm.

* usage: specifies the amount of memory used in percentage.

* threshold: The threshold triggering the alarm in percentage.

* severity: The severity level of the message. There are four levels, i.e., critical, high, middle, and low.

* message: Simple information as a human readable text string such as "The memory usage exceeded the threshold" or with extra information.

6.1.2. CPU Alarm

CPU is the Central Processing Unit that executes basic operations of the system. The cpu-alarm is emitted when the CPU usage exceeds the threshold. The following information should be included in a CPU Alarm:

* event-name: cpu-alarm.

* usage: Specifies the CPU utilization in percentage.

* threshold: The threshold triggering the event in percentage.

* severity: The severity level of the message. There are four levels, i.e., critical, high, middle, and low.

* message: Simple information as a human readable text string such as "The CPU usage exceeded the threshold" or with extra information.

6.1.3. Disk (Storage) Alarm

Disk or storage is the hardware to store information for a long time, i.e., Hard Disk or Solid-State Drive. The disk-alarm is emitted when the Disk usage exceeds the threshold. The following information should be included in a Disk Alarm:

* event-name: disk-alarm.

* usage: Specifies the ratio of the used disk space to the whole disk space in terms of percentage.

* threshold: The threshold triggering the event in percentage.

* severity: The severity level of the message. There are four levels, i.e., critical, high, middle, and low.

* message: Simple information as a human readable text string such as "The disk usage exceeded the threshold" or with extra information.

6.1.4. Hardware Alarm

The hardware-alarm is emitted when a hardware, e.g., CPU, memory, disk, or interface, problem is detected. The following information should be included in a Hardware Alarm:

* event-name: hardware-alarm.

* component-name: It indicates the hardware component responsible for generating this alarm.

* severity: The severity level of the message. There are four levels, i.e., critical, high, middle, and low.

* message: Simple information as a human readable text string such as "The hardware component has failed or degraded" or with extra information.

6.1.5. Interface Alarm

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. The following information should be included in an Interface Alarm:

* event-name: interface-alarm.

* interface-name: The name of the interface.

* interface-state: The status of the interface, i.e., down, up (not congested), congested (up but congested), testing, unknown, dormant, not-present, and lower-layer-down.

* severity: The severity level of the message. There are four levels, i.e., critical, high, middle, and low.

* message: Simple information as a human readable text string such as "The interface is 'interface-state'" or with extra information.

6.2. System Events

System events (as alerts) have the following characteristics:

* acquisition-method: subscription

* emission-type: on-change

* dampening-type: on-repetition or no-dampening

6.2.1. Access Violation

The 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. The following information should be included in this event:

* event-name: access-violation.

* identity: The information to identify the attempted access violation. The minimum information (extensible) that should be included:

1. user: The unique username that attempted access violation.

2. group: Group(s) to which a user belongs. A user can belong to multiple groups.

3. ip-address: The IP address of the user that triggered the event.

4. l4-port-number: The transport layer port number used by the user.

* authentication: The method to verify the valid user, i.e., pre-configured-key and certificate-authority.

* message: The message as a human readable text string to give the context of the event, such as "Access is denied".

6.2.2. Configuration Change

A configuration change is a system event when a new configuration is added or an existing configuration is modified. The following information should be included in this event:

* event-name: configuration-change.

* identity: The information to identify the user that updated the configuration. The minimum information (extensible) that should be included:

  1. user: The unique username that changes the configuration.

  2. group: Group(s) to which a user belongs. A user can belong to multiple groups.

  3. ip-address: The IP address of the user that triggered the event.

  4. l4-port-number: The transport layer port number used by the user.

* authentication: The method to verify the valid user, i.e., pre-configured-key and certificate-authority.

* message: The message as a human readable text string to give the context of the event, such as "Configuration is modified", "New configuration is added", or "A configuration has been removed".

* changes: Describes the modification that was made to the configuration. The minimum information that must be provided is the name of the policy that has been altered (added, modified, or removed). Other detailed information about the configuration changes is up to the implementation.

6.2.3. Session Table Event

A session is defined as a connection (i.e., traffic flow) of a data plane (e.g., TCP, UDP, and SCTP). Session Table Event is the event triggered by the session table of an NSF. A session table holds the information of the currently active sessions. The following information should be included in a Session Table Event:

* event-name: detection-session-table.

* current-session: The number of concurrent sessions.

* maximum-session: The maximum number of sessions that the session table can support.

* threshold: The threshold (in terms of an allowed number of sessions) triggering the event.

* message: The message as a human readable text string to give the context of the event, such as "The number of sessions exceeded the table threshold".

6.2.4. Traffic Flows

Traffic flows need to be monitored because they might be used for security attacks to the network. The following information should be included in this event:

* event-name: traffic-flows.

* interface-name: The mnemonic name of the network interface

* interface-type: The type of a network interface such as an ingress or egress interface.

* src-mac: The source MAC address of the traffic flow. This information may or may not be included depending on the type of traffic flow. For example, the information will be useful and should be included if the traffic flows are traffic flows of Link Layer Discovery Protocol (LLDP) [IEEE-802.1AB], Address Resolution Protocol (ARP) for IPv4 [RFC0826], and Neighbor Discovery Protocol (ND) for IPv6 [RFC4861].

* dst-mac: The destination MAC address of the traffic flow. This information may or may not be included depending on the type of traffic flow. For example, the information will be useful and should be included if the traffic flows are LLDP, ARP for IPv4, or ND for IPv6 traffic flows.

* src-ip: The source IPv4 or IPv6 address of the traffic flow.

* dst-ip: The destination IPv4 or IPv6 address of the traffic flow.

* src-port: The transport layer source port number of the traffic flow.

* dst-port: The transport layer destination port number of the traffic flow.

* protocol: The protocol of the traffic flow.

* measurement-time: The duration of the measurement in seconds for the arrival rate and arrival throughput of packets of a traffic flow. These two metrics (i.e., arrival rate and arrival throughput) are measured over the past measurement duration before now.

* arrival-rate: Arrival rate of packets of the traffic flow in packets per second measured over the past "measurement-time".

* arrival-throughput: Arrival rate of packets of the traffic flow in bytes per second measured over the past "measurement-time".

Note that the NSF Monitoring Interface data model is focused on a generic method to collect the monitoring information of systems and NSFs including traffic flows related to security attacks and system resource usages. On the other hand, IPFIX [RFC7011] is a standard method to collect general information on traffic flows rather than security.

6.3. NSF Events

The NSF events provide the event that is detected by a specific NSF that supported a certain capability. This section only discusses the monitoring data for the advanced NSFs discussed in [I-D.ietf-i2nsf-capability-data-model]. The NSF events information can be extended to support other types of NSF. NSF events have the following characteristics:

* acquisition-method: subscription

* emission-type: on-change

* dampening-type: on-repetition or no-dampening

6.3.1. DDoS Detection

The following information should be included in a Denial-of-Service (DoS) or Distributed Denial-of-Service (DDoS) Event:

* event-name: detection-ddos.

* attack-type: The type of DoS or DDoS Attack, i.e., SYN flood, ACK flood, SYN-ACK flood, FIN/RST flood, TCP Connection flood, UDP flood, ICMP flood, HTTPS flood, HTTP flood, DNS query flood, DNS reply flood, SIP flood, TLS flood, and NTP amplification flood. This can be extended with additional types of DoS or DDoS attack.

* attack-src-ip: The IP addresses of the source of the DDoS attack. Note that not all IP addresses should 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.

* attack-dst-ip: The destination IPv4 or IPv6 addresses of attack traffic. It can hold multiple IPv4 or IPv6 addresses.

* attack-src-port: The transport layer source port numbers of the attack traffic. Note that not all ports will have been seen on all the corresponding source IP addresses.

* attack-dst-port: The transport layer destination port numbers that the attack traffic aims at. Note that not all ports will have been seen on all the corresponding destination IP addresses.

* start-time: The time stamp indicating when the attack started. The time format used is following the rules in Section 5.6 of [RFC3339].

* end-time: The time stamp indicating when the attack ended. If the attack is still ongoing when sending out the notification, this field can be empty. The time format used is following the rules in Section 5.6 of [RFC3339].

* attack-rate: The packets per second of attack traffic.

* attack-throughput: The bytes per second of attack traffic.

* rule-name: The name of the I2NSF Policy Rule being triggered. Note that rule-name is used to match a detected NSF event with a policy rule in [I-D.ietf-i2nsf-nsf-facing-interface-dm].

6.3.2. Virus Event

This information is used when a virus is detected within a traffic flow or inside a host. Note that "malware" is a more generic word for malicious software, including virus and worm. In the document, "virus" is used to represent "malware" such that they are interchangeable. The following information should be included in a Virus Event:

* event-name: detection-virus.

* virus-name: Name of the virus.

* virus-type: Type of the virus. e.g., trojan, worm, and macro virus.

* The following information is used only when the virus is detected within the traffic flow and not yet attacking the host:

  - dst-ip: The destination IP address of the flow where the virus is found.

  - src-ip: The source IP address of the flow where the virus is found.

  - src-port: The source port of the flow where the virus is found.

  - dst-port: The destination port of the flow where the virus is found.

* The following information is used only when the virus is detected within a host system:

  - host: The name or IP address of 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.

  - os: The operating system of the host that has the virus.

  - file-type: The type of file (indicated by the file’s suffix, e.g., .exe) virus code is found in (if applicable).

- file-name: The name of the file where the virus is hidden.

* rule-name: The name of the rule being triggered.

Note "host" is used only when the virus is detected within a host itself. Thus, the traffic flow information such as the source and destination IP addresses is not important, so the elements of the traffic flow (i.e., dst-ip, src-ip, src-port, and dst-port) are not specified above. On the other hand, when the virus is detected within a traffic flow and not yet attacking a host, the element of "host" is not specified above.

6.3.3. Intrusion Event

The following information should be included in an Intrusion Event:

* event-name: detection-intrusion.

* attack-type: Attack type, e.g., brutal force or buffer overflow.

* src-ip: The source IP address of the flow.

* dst-ip: The destination IP address of the flow.

* src-port: The source port number of the flow.

* dst-port: The destination port number of the flow

* protocol: The employed transport layer protocol. e.g., TCP or UDP. 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 generic UDP traffic and will be considered in the future I2NSF documents.

* app: The employed application layer protocol. e.g., HTTP or FTP.

* rule-name: The name of the I2NSF Policy Rule being triggered.

6.3.4. Web Attack Event

The following information should be included in a Web Attack Alarm:

* event-name: detection-web-attack.

* attack-type: Concrete web attack type. e.g., SQL injection, command injection, XSS, or CSRF.

* src-ip: The source IP address of the packet.

* dst-ip: The destination IP address of the packet.

* src-port: The source port number of the packet.

* dst-port: The destination port number of the packet.

* req-method: The HTTP method of the request. For instance, "PUT" and "GET" in HTTP.

* req-target: The HTTP Request Target.

* response-code: The HTTP Response status code.

* cookies: The HTTP Cookie header field of the request from the user agent. Note that though cookies have many historical infelicities that degrade security and privacy, the Cookie and Set-Cookie header fields are widely used on the Internet [RFC6265]. Thus, the cookies 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.

* req-host: The HTTP Host header field of the request.

* filtering-type: URL filtering type. e.g., deny-list, allow-list, and unknown.

* rule-name: The name of the I2NSF Policy Rule being triggered.

6.3.5. VoIP/VoCN Event

The following information should be included in a VoIP (Voice over Internet Protocol) and VoCN (Voice over Cellular Network, such as Voice over LTE or 5G) Event:

* event-name: detection-voip-vocn

* source-voice-id: The detected source voice Call ID for VoIP and VoCN that violates the policy.

* destination-voice-id: The destination voice Call ID for VoIP and VoCN that violates the policy.

* user-agent: The user agent for VoIP and VoCN that violates the policy.

* src-ip: The source IP address of the VoIP/VoCN.

* dst-ip: The destination IP address of the VoIP/VoCN.

* src-port: The source port number of the VoIP/VoCN.

* dst-port: The destination port number of VoIP/VoCN.

* rule-name: The name of the I2NSF Policy Rule being triggered.

6.4. System Logs

System log is a record that is used to monitor the activity of the user on the NSF and the status of the NSF. System logs have the following characteristics:

* acquisition-method: subscription or query

* emission-type: on-change or periodic

* dampening-type: on-repetition or no-dampening

6.4.1. Access Log

Access logs record administrators' login, logout, and operations on a device. By analyzing them, some security vulnerabilities can be identified. The following information should be included in an operation report:

* identity: The information to identify the user. The minimum information (extensible) that should be included:

  1. user: The unique username that attempted access violation.

  2. group: Group(s) to which a user belongs. A user can belong to multiple groups.

  3. ip-address: The IP address of the user that triggered the event.

  4. l4-port-number: The transport layer port number used by the user.

* authentication: The method to verify the valid user, i.e., pre-configured-key and certificate-authority.

* operation-type: The operation type that the administrator executed, e.g., login, logout, configuration, and other.

* input: The operation performed by a user after login. The operation is a command given by a user.

* output: The result after executing the input.

6.4.2. Resource Utilization Log

Running reports record the device system's running status, which is useful for device monitoring. The following information should be included in running report:

* system-status: The current system's running status.

* cpu-usage: Specifies the aggregated CPU usage in percentage.

* memory-usage: Specifies the memory usage in percentage.

* disk-id: Specifies the disk ID to identify the storage disk.

* disk-usage: Specifies the disk usage of disk-id in percentage.

* disk-space-left: Specifies the available disk space left of disk-id in percentage.

* session-number: Specifies total concurrent sessions.

* process-number: Specifies total number of systems processes.

* interface-id: Specifies the interface ID to identify the network interface.

* in-traffic-rate: The total inbound data plane traffic rate in packets per second.

* out-traffic-rate: The total outbound data plane traffic rate in packets per second.

* in-traffic-throughput: The total inbound data plane traffic throughput in bytes per second.

* out-traffic-throughput: The total outbound data plane traffic throughput in bytes per second.

Note that "traffic" includes only the data plane since the monitoring interface focuses on the monitoring of traffic flows for applications, rather than the control plane. In the document, "packet" includes a layer-2 frame, so "packet" and "frame" are interchangeable. Also, note that system resources (e.g., CPU,

memory, disk, and interface) are monitored for the sake of security in NSFs even though they are common ones to be monitored by a generic Operations, Administration and Maintenance (OAM) protocol (or module).

6.4.3. User Activity Log

User activity logs provide visibility into users' online records (such as login time, online/lockout duration, and login IP addresses) and the actions that users perform. User activity reports are helpful to identify exceptions during a user's login and network access activities. This information should be included in a user's activity report:

* identity: The information to identify the user. The minimum information (extensible) that should be included is as follows:

  1. user: The unique username that attempted access violation.

  2. group: Group(s) to which a user belongs. A user can belong to multiple groups.

  3. ip-address: The IP address of the user that triggered the event.

  4. l4-port-number: The transport layer port number used by the user.

* authentication: The method to verify the valid user, i.e., pre-configured-key and certificate-authority.

* online-duration: The duration of a user's activeness (stays in login) during a session.

* logout-duration: The duration of a user's inactiveness (not in login) from the last session.

* additional-info: Additional Information for login:

  1. type: User activities. e.g., Successful User Login, Failed Login attempts, User Logout, Successful User Password Change, Failed User Password Change, User Lockout, and User Unlocking.

  2. cause: Cause of a failed user activity.

6.5. NSF Logs

NSF logs have the folowing characteristics:

* acquisition-method: subscription or query

* emission-type: on-change

* dampening-type: on-repetition or no-dampening

6.5.1. Deep Packet Inspection Log

Deep Packet Inspection (DPI) Logs provide statistics of transit traffic at an NSF such that the traffic includes uploaded and downloaded files/data, sent/received emails, and blocking/alert records on websites. It is helpful to learn risky user behaviors and why access to some URLs is blocked or allowed with an alert record.

* attack-type: DPI action types. e.g., File Blocking, Data Filtering, and Application Behavior Control.

* src-ip: The source IP address of the flow.

* dst-ip: The destination IP address of the flow.

* src-port: The source port number of the flow.

* dst-port: The destination port number of the flow

* rule-name: The name of the I2NSF Policy Rule being triggered.

* action: Action defined in the file blocking rule, data filtering rule, or application behavior control rule that traffic matches.

6.6. System Counter

System counter has the following characteristics:

* acquisition-method: subscription or query

* emission-type: periodic

* dampening-type: no-dampening

6.6.1. Interface Counter

Interface counters provide visibility into traffic into and out of an NSF, and bandwidth usage.

* interface-name: Network interface name configured in NSF.

* protocol: The type of network protocol (e.g., IPv4, IPv6, TCP, and UDP). If this field is empty, then the counter is used for all protocols.

* measurement-time: The duration of the measurement in seconds for the calculation of statistics such as traffic rate and throughput. The statistic attributes are measured over the past measurement duration before now.

* in-total-traffic-pkts: Total inbound packets.

* out-total-traffic-pkts: Total outbound packets.

* in-total-traffic-bytes: Total inbound bytes.

* out-total-traffic-bytes: Total outbound bytes.

* in-drop-traffic-pkts: Total inbound drop packets caused by a policy or hardware/resource error.

* out-drop-traffic-pkts: Total outbound drop packets caused by a policy or hardware/resource error.

* in-drop-traffic-bytes: Total inbound drop bytes caused by a policy or hardware/resource error.

* out-drop-traffic-bytes: Total outbound drop bytes caused by a policy or hardware/resource error.

* total-traffic: The total number of traffic packets (in and out) in the NSF.

* in-traffic-average-rate: Inbound traffic average rate in packets per second.

* in-traffic-peak-rate: Inbound traffic peak rate in packets per second.

* in-traffic-average-throughput: Inbound traffic average throughput in bytes per second.

* in-traffic-peak-throughput: Inbound traffic peak throughput in bytes per second.

* out-traffic-average-rate: Outbound traffic average rate in packets per second.

* out-traffic-peak-rate: Outbound traffic peak rate in packets per second.

* out-traffic-average-throughput: Outbound traffic average throughput in bytes per second.

* out-traffic-peak-throughput: Outbound traffic peak throughput in bytes per second.

* discontinuity-time: The time of the most recent occasion at which any one or more of the counters suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this node contains the time the local management subsystem was re-initialized. The time format used is following the rules in Section 5.6 of [RFC3339].

6.7. NSF Counters

NSF counters have the following characteristics:

* acquisition-method: subscription or query

* emission-type: periodic

* dampening-type: no-dampening

6.7.1. Firewall Counter

Firewall counters provide visibility into traffic signatures and bandwidth usage that correspond to the policy that is configured in a firewall.

* policy-name: Security policy name that traffic matches.

* measurement-time: The duration of the measurement in seconds for the calculation of statistics such as traffic rate and throughput. The statistic attributes are measured over the past measurement duration before now.

* in-interface: Inbound interface of traffic.

* out-interface: Outbound interface of traffic.

* total-traffic: The total number of traffic packets (in and out) in the firewall.

* in-traffic-average-rate: Inbound traffic average rate in packets per second.

* in-traffic-peak-rate: Inbound traffic peak rate in packets per second.

* in-traffic-average-throughput: Inbound traffic average throughput in bytes per second.

* in-traffic-peak-throughput: Inbound traffic peak throughput in bytes per second.

* out-traffic-average-rate: Outbound traffic average rate in packets per second.

* out-traffic-peak-rate: Outbound traffic peak rate in packets per second.

* out-traffic-average-throughput: Outbound traffic average throughput in bytes per second.

* out-traffic-peak-throughput: Outbound traffic peak throughput in bytes per second.

* discontinuity-time: The time on the most recent occasion at which any one or more of the counters suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this node contains the time the local management subsystem was re-initialized. The time format used is following the rules in Section 5.6 of [RFC3339].

6.7.2. Policy Hit Counter

Policy hit counters record the security policy that traffic matches and its hit count. That is, when a packet actually matches a policy, it should be added to the statistics of a "policy hit counter" of the policy. The "policy hit counter" provides the "policy-name" that matches the policy’s name in the NSF-Facing Interface YANG data model [I-D.ietf-i2nsf-nsf-facing-interface-dm]. It can check if policy configurations are correct or not.

* policy-name: Security policy name that traffic matches.

* hit-times: The number of times that the security policy matches the specified traffic.

* discontinuity-time: The time on the most recent occasion at which any one or more of the counters suffered a discontinuity. If no such discontinuities have occurred since the last re-initialization of the local management subsystem, then this node contains the time the local management subsystem was re-initialized. The time format used is following the rules in Section 5.6 of [RFC3339].

7. YANG Tree Structure of NSF Monitoring YANG Module

The tree structure of the NSF monitoring YANG module is provided below:

```
module: ietf-i2nsf-monitoring-interface
  +--ro i2nsf-counters
     |  +--ro vendor-name?          string
     |  +--ro device-model?         string
     |  +--ro software-version?     string
     |  +--ro nsf-name              union
     |  +--ro timestamp?            yang:date-and-time
     |  +--ro acquisition-method?   identityref
     |  +--ro emission-type?        identityref
     |  +--ro system-interface* [interface-name]
     |  |  +--ro interface-name                    if:interface-ref
     |  |  +--ro protocol?                         identityref
     |  |  +--ro in-total-traffic-pkts?            yang:counter64
     |  |  +--ro out-total-traffic-pkts?           yang:counter64
     |  |  +--ro in-total-traffic-bytes?           uint64
     |  |  +--ro out-total-traffic-bytes?          uint64
     |  |  +--ro in-drop-traffic-pkts?             yang:counter64
     |  |  +--ro out-drop-traffic-pkts?            yang:counter64
     |  |  +--ro in-drop-traffic-bytes?            uint64
     |  |  +--ro out-drop-traffic-bytes?           uint64
     |  |  +--ro discontinuity-time                yang:date-and-time
     |  |  +--ro measurement-time?                 uint32
     |  |  +--ro total-traffic?                    yang:counter64
     |  |  +--ro in-traffic-average-rate?          uint64
     |  |  +--ro in-traffic-peak-rate?             uint64
     |  |  +--ro in-traffic-average-throughput?    uint64
     |  |  +--ro in-traffic-peak-throughput?       uint64
     |  |  +--ro out-traffic-average-rate?         uint64
     |  |  +--ro out-traffic-peak-rate?            uint64
     |  |  +--ro out-traffic-average-throughput?   uint64
     |  |  +--ro out-traffic-peak-throughput?      uint64
     |  +--ro nsf-firewall* [policy-name]
     |  |  +--ro in-interface?                     if:interface-ref
     |  |  +--ro out-interface?                    if:interface-ref
     |  |  +--ro policy-name     -> /i2nsfnfi:i2nsf-security-policy/name
```

```
     |  |  +--ro discontinuity-time                yang:date-and-time
     |  |  +--ro measurement-time?                 uint32
     |  |  +--ro total-traffic?                    yang:counter64
     |  |  +--ro in-traffic-average-rate?          uint64
     |  |  +--ro in-traffic-peak-rate?             uint64
     |  |  +--ro in-traffic-average-throughput?    uint64
     |  |  +--ro in-traffic-peak-throughput?       uint64
     |  |  +--ro out-traffic-average-rate?         uint64
     |  |  +--ro out-traffic-peak-rate?            uint64
     |  |  +--ro out-traffic-average-throughput?   uint64
     |  |  +--ro out-traffic-peak-throughput?      uint64
     |  +--ro nsf-policy-hits* [policy-name]
     |     +--ro policy-name     -> /i2nsfnfi:i2nsf-security-policy/name
     |     +--ro discontinuity-time    yang:date-and-time
     |     +--ro hit-times?            yang:counter64
     +--rw i2nsf-monitoring-configuration
        +--rw i2nsf-system-detection-alarm
        |  +--rw enabled?        boolean
        |  +--rw system-alarm* [alarm-type]
        |     +--rw alarm-type          enumeration
        |     +--rw threshold?          uint8
        |     +--rw dampening-period?   centiseconds
        +--rw i2nsf-system-detection-event
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-traffic-flows
        |  +--rw dampening-period?   centiseconds
        |  +--rw enabled?            boolean
        +--rw i2nsf-nsf-detection-ddos {i2nsf-nsf-detection-ddos}?
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-detection-virus {i2nsf-nsf-detection-virus}?
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-detection-session-table
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-detection-intrusion
                                      {i2nsf-nsf-detection-intrusion}?
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-detection-web-attack
                                     {i2nsf-nsf-detection-web-attack}?
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-detection-voip-vocn
                                      {i2nsf-nsf-detection-voip-vocn}?
        |  +--rw enabled?            boolean
```

```
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-system-access-log
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-system-res-util-log
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-system-user-activity-log
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-nsf-log-dpi {i2nsf-nsf-log-dpi}?
        |  +--rw enabled?            boolean
        |  +--rw dampening-period?   centiseconds
        +--rw i2nsf-counter
           +--rw period?   uint16

     notifications:
       +---n i2nsf-event
       |  +--ro vendor-name?                              string
       |  +--ro device-model?                             string
       |  +--ro software-version?                         string
       |  +--ro nsf-name                                  union
       |  +--ro message?                                  string
       |  +--ro language?                                 string
       |  +--ro acquisition-method?                       identityref
       |  +--ro emission-type?                            identityref
       |  +--ro dampening-type?                           identityref
       |  +--ro (sub-event-type)?
       |     +--:(i2nsf-system-detection-alarm)
       |     |  +--ro i2nsf-system-detection-alarm
       |     |     +--ro alarm-category?    identityref
       |     |     +--ro component-name?    string
       |     |     +--ro interface-name?    if:interface-ref
       |     |     +--ro interface-state?   enumeration
       |     |     +--ro severity?          severity
       |     |     +--ro usage?             uint8
       |     |     +--ro threshold?         uint8
       |     +--:(i2nsf-system-detection-event)
       |     |  +--ro i2nsf-system-detection-event
       |     |     +--ro event-category?   identityref
       |     |     +--ro user              string
       |     |     +--ro group*            string
       |     |     +--ro ip-address        inet:ip-address-no-zone
       |     |     +--ro l4-port-number    inet:port-number
       |     |     +--ro authentication?   identityref
       |     |     +--ro changes* [policy-name]
       |     |        +--ro policy-name
       |                             -> /i2nsfnfi:i2nsf-security-policy/name
```

```
        |      +--:(i2nsf-traffic-flows)
        |      |  +--ro i2nsf-traffic-flows
        |      |     +--ro interface-name?       if:interface-ref
        |      |     +--ro interface-type?       enumeration
        |      |     +--ro src-mac?              yang:mac-address
        |      |     +--ro dst-mac?              yang:mac-address
        |      |     +--ro src-ip?               inet:ip-address-no-zone
        |      |     +--ro dst-ip?               inet:ip-address-no-zone
        |      |     +--ro protocol?             identityref
        |      |     +--ro src-port?             inet:port-number
        |      |     +--ro dst-port?             inet:port-number
        |      |     +--ro measurement-time?     uint32
        |      |     +--ro arrival-rate?         uint64
        |      |     +--ro arrival-throughput?   uint64
        |      +--:(i2nsf-nsf-detection-session-table)
        |         +--ro i2nsf-nsf-detection-session-table
        |            +--ro current-session?   uint32
        |            +--ro maximum-session?   uint32
        |            +--ro threshold?         uint32
        +---n i2nsf-log
        |  +--ro vendor-name?                           string
        |  +--ro device-model?                          string
        |  +--ro software-version?                      string
        |  +--ro nsf-name                               union
        |  +--ro message?                               string
        |  +--ro language?                              string
        |  +--ro acquisition-method?                    identityref
        |  +--ro emission-type?                         identityref
        |  +--ro dampening-type?                        identityref
        |  +--ro (sub-logs-type)?
        |     +--:(i2nsf-nsf-system-access-log)
        |     |  +--ro i2nsf-nsf-system-access-log
        |     |     +--ro user              string
        |     |     +--ro group*            string
        |     |     +--ro ip-address        inet:ip-address-no-zone
        |     |     +--ro l4-port-number    inet:port-number
        |     |     +--ro authentication?   identityref
        |     |     +--ro operation-type?   operation-type
        |     |     +--ro input?            string
        |     |     +--ro output?           string
        |     +--:(i2nsf-system-res-util-log)
        |     |  +--ro i2nsf-system-res-util-log
        |     |     +--ro system-status?   enumeration
        |     |     +--ro cpu-usage?       uint8
        |     |     +--ro memory-usage?    uint8
        |     |     +--ro disks* [disk-id]
        |     |     |  +--ro disk-id            string
        |     |     |  +--ro disk-usage?        uint8
```

```
        |      |      |  +--ro disk-space-left?   uint8
        |      |      +--ro session-num?     uint32
        |      |      +--ro process-num?     uint32
        |      |      +--ro interface* [interface-id]
        |      |         +--ro interface-id              string
        |      |         +--ro in-traffic-rate?          uint64
        |      |         +--ro out-traffic-rate?         uint64
        |      |         +--ro in-traffic-throughput?    uint64
        |      |         +--ro out-traffic-throughput?   uint64
        |      +--:(i2nsf-system-user-activity-log)
        |      |  +--ro i2nsf-system-user-activity-log
        |      |     +--ro user               string
        |      |     +--ro group*             string
        |      |     +--ro ip-address         inet:ip-address-no-zone
        |      |     +--ro l4-port-number     inet:port-number
        |      |     +--ro authentication?    identityref
        |      |     +--ro online-duration?   uint32
        |      |     +--ro logout-duration?   uint32
        |      |     +--ro additional-info
        |      |        +--ro type?    enumeration
        |      |        +--ro cause?   string
        |      +--:(i2nsf-nsf-log-dpi) {i2nsf-nsf-log-dpi}?
        |         +--ro i2nsf-nsf-log-dpi
        |            +--ro attack-type?   identityref
        |            +--ro src-ip?        inet:ip-address-no-zone
        |            +--ro src-port?      inet:port-number
        |            +--ro dst-ip?        inet:ip-address-no-zone
        |            +--ro dst-port?      inet:port-number
        |            +--ro rule-name
        |                    -> /i2nsfnfi:i2nsf-security-policy/rules/name
        |            +--ro action*        identityref
        +---n i2nsf-nsf-event
           +--ro vendor-name?                           string
           +--ro device-model?                          string
           +--ro software-version?                      string
           +--ro nsf-name                               union
           +--ro message?                               string
           +--ro language?                              string
           +--ro acquisition-method?                    identityref
           +--ro emission-type?                         identityref
           +--ro dampening-type?                        identityref
           +--ro (sub-event-type)?
              +--:(i2nsf-nsf-detection-ddos) {i2nsf-nsf-detection-ddos}?
              |  +--ro i2nsf-nsf-detection-ddos
              |     +--ro attack-type?         identityref
              |     +--ro start-time           yang:date-and-time
              |     +--ro end-time?            yang:date-and-time
              |     +--ro attack-src-ip*       inet:ip-address-no-zone
```

```
             |      +--ro attack-dst-ip*       inet:ip-address-no-zone
             |      +--ro attack-src-port*     inet:port-number
             |      +--ro attack-dst-port*     inet:port-number
             |      +--ro rule-name
             |              -> /i2nsfnfi:i2nsf-security-policy/rules/name
             |      +--ro attack-rate?         uint64
             |      +--ro attack-throughput?   uint64
             +--:(i2nsf-nsf-detection-virus)
             |                                {i2nsf-nsf-detection-virus}?
             |  +--ro i2nsf-nsf-detection-virus
             |      +--ro src-ip?       inet:ip-address-no-zone
             |      +--ro src-port?     inet:port-number
             |      +--ro dst-ip?       inet:ip-address-no-zone
             |      +--ro dst-port?     inet:port-number
             |      +--ro rule-name
             |              -> /i2nsfnfi:i2nsf-security-policy/rules/name
             |      +--ro virus-name?   string
             |      +--ro virus-type?   identityref
             |      +--ro host?         union
             |      +--ro file-type?    string
             |      +--ro file-name?    string
             |      +--ro os?           string
             +--:(i2nsf-nsf-detection-intrusion)
             |                           {i2nsf-nsf-detection-intrusion}?
             |  +--ro i2nsf-nsf-detection-intrusion
             |      +--ro src-ip?        inet:ip-address-no-zone
             |      +--ro src-port?      inet:port-number
             |      +--ro dst-ip?        inet:ip-address-no-zone
             |      +--ro dst-port?      inet:port-number
             |      +--ro rule-name
             |              -> /i2nsfnfi:i2nsf-security-policy/rules/name
             |      +--ro protocol?      identityref
             |      +--ro app?           identityref
             |      +--ro attack-type?   identityref
             +--:(i2nsf-nsf-detection-web-attack)
             |                          {i2nsf-nsf-detection-web-attack}?
             |  +--ro i2nsf-nsf-detection-web-attack
             |      +--ro src-ip?           inet:ip-address-no-zone
             |      +--ro src-port?         inet:port-number
             |      +--ro dst-ip?           inet:ip-address-no-zone
             |      +--ro dst-port?         inet:port-number
             |      +--ro rule-name
             |              -> /i2nsfnfi:i2nsf-security-policy/rules/name
             |      +--ro attack-type?      identityref
             |      +--ro req-method?       identityref
             |      +--ro req-target?       string
             |      +--ro filtering-type*   identityref
             |      +--ro cookies?          string
```

```
             |     +--ro req-host?         string
             |     +--ro response-code?    string
             +--:(i2nsf-nsf-detection-voip-vocn)
                                       {i2nsf-nsf-detection-voip-vocn}?
                +--ro i2nsf-nsf-detection-voip-vocn
                   +--ro src-ip?                 inet:ip-address-no-zone
                   +--ro src-port?               inet:port-number
                   +--ro dst-ip?                 inet:ip-address-no-zone
                   +--ro dst-port?               inet:port-number
                   +--ro rule-name
                           -> /i2nsfnfi:i2nsf-security-policy/rules/name
                   +--ro source-voice-id*        string
                   +--ro destination-voice-id*   string
                   +--ro user-agent*             string
```

Figure 1: NSF Monitoring YANG Module Tree

8. YANG Data Model of NSF Monitoring YANG Module

This section describes a YANG module of I2NSF NSF Monitoring. The data model provided in this document uses identities to be used to get information of the monitored of an NSF's monitoring data. Every identity used in the document gives information or status about the current situation of an NSF. This YANG module imports from [RFC6991], [RFC8343], and [I-D.ietf-i2nsf-nsf-facing-interface-dm], and makes references to [RFC0768] [RFC0791] [RFC0792] [RFC0826] [RFC0854] [RFC1939] [RFC0959] [RFC2595] [RFC4340] [RFC4443] [RFC4861] [RFC5321] [RFC5646] [RFC6242] [RFC6265] [RFC8200] [RFC8641] [RFC9051] [I-D.ietf-httpbis-http2bis] [I-D.ietf-httpbis-messaging] [I-D.ietf-httpbis-semantics] [I-D.ietf-tcpm-rfc793bis] [I-D.ietf-tsvwg-rfc4960-bis] [IANA-HTTP-Status-Code] [IEEE-802.1AB]

```
<CODE BEGINS> file "ietf-i2nsf-monitoring-interface@2022-06-01.yang"
module ietf-i2nsf-monitoring-interface {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface";
  prefix
    i2nsfmi;
  import ietf-inet-types {
    prefix inet;
    reference
      "Section 4 of RFC 6991";
  }
  import ietf-yang-types {
    prefix yang;
    reference
```

```
        "Section 3 of RFC 6991";
    }
    import ietf-i2nsf-nsf-facing-interface {
      prefix i2nsfnfi;
      reference
        "Section 4.1 of draft-ietf-i2nsf-nsf-facing-interface-dm-29";
    }
    import ietf-interfaces {
      prefix if;
      reference
        "Section 5 of RFC 8343";
    }
    organization
      "IETF I2NSF (Interface to Network Security Functions)
       Working Group";
    contact
      "WG Web: <https://datatracker.ietf.org/wg/i2nsf>
       WG List: <mailto:i2nsf@ietf.org>

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

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

    description
      "This module is a YANG module for I2NSF NSF Monitoring.

       The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
       'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
       'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this
       document are to be interpreted as described in BCP 14
       (RFC 2119) (RFC 8174) when, and only when, they appear
       in all capitals, as shown here.

       Copyright (c) 2022 IETF Trust and the persons identified as
       authors of the code.  All rights reserved.

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

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

```
     revision "2022-06-01" {
       description "Latest revision";
       reference
         "RFC XXXX: I2NSF NSF Monitoring Interface YANG Data Model";

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

     /*
      * Typedefs
      */

     typedef severity {
       type enumeration {
         enum critical {
           description
             "The 'critical' severity level indicates that
              an immediate corrective action is required.
              A 'critical' severity is reported when a service
              becomes totally out of service and must be restored.";
         }
         enum high {
           description
             "The 'high' severity level indicates that
              an urgent corrective action is required.
              A 'high' severity is reported when there is
              a severe degradation in the capability of the
              service and its full capability must be restored.";
         }
         enum middle {
           description
             "The 'middle' severity level indicates the
              existence of a non-service-affecting fault
              condition and corrective action should be done
              to prevent a more serious fault. The 'middle'
              severity is reported when the detected problem
              is not degrading the capability of the service, but
              some service degradation might happen if not
              prevented.";
         }
         enum low {
           description
             "The 'low' severity level indicates the detection
              of a potential fault before any effect is observed.
              The 'low' severity is reported when an action should
              be done before a fault happen.";
         }
```

```
      }
      description
        "An indicator representing severity levels. The severity
         levels starting from the highest are critical, high, middle,
         and low.";
    }

    typedef operation-type {
      type enumeration {
        enum login {
          description
            "The operation type is Login.";
        }
        enum logout {
          description
            "The operation type is Logout.";
        }
        enum configuration {
          description
            "The operation type is Configuration. The configuration
             operation includes the command for writing a new
             configuration and modifying an existing configuration.";
        }
        enum other {
          description
            "The operation type is Other operation. This other
             includes all operations done by a user except login,
             logout, and configuration.";
        }
      }
      description
        "The type of operation done by a user during a session.
         The user operation is not considering their privileges.";
    }

    typedef login-role {
      type enumeration {
        enum administrator {
          description
            "Administrator (i.e., Superuser)'s login role.
             Non-restricted role.";
        }
        enum user {
          description
            "User login role. Semi-restricted role, some data and
             configurations are available but confidential or important
             data and configuration are restricted.";
        }
```

```
        enum guest {
          description
            "Guest login role. Restricted role, only few read data are
             available and write configurations are restricted.";
        }
      }
      description
        "The privilege level of the user account.";
    }

    typedef centiseconds {
      type uint32;
      description
        "A period of time, measured in units of 0.01 seconds.";
    }

    /*
     * Identity
     */

    identity characteristics {
      description
        "Base identity for monitoring information
         characteristics";
    }
    identity acquisition-method {
      base characteristics;
      description
        "The type of acquisition-method. It can be multiple
         types at once.";
    }
    identity subscription {
      base acquisition-method;
      description
        "The acquisition-method type is subscription.";
    }
    identity query {
      base acquisition-method;
      description
        "The acquisition-method type is query.";
    }
    identity emission-type {
      base characteristics;
      description
        "The type of emission-type.";
    }
    identity periodic {
      base emission-type;
```

```
      description
        "The emission-type type is periodic.";
    }
    identity on-change {
      base emission-type;
      description
        "The emission-type type is on-change.";
    }
    identity dampening-type {
      base characteristics;
      description
        "The type of message dampening to stop the rapid transmission
         of messages, such as on-repetition and no-dampening.";
    }
    identity no-dampening {
      base dampening-type;
      description
        "The dampening-type is no-dampening. No-dampening type does
         not limit the transmission for the messages of the same
         type.";
    }
    identity on-repetition {
      base dampening-type;
      description
        "The dampening-type is on-repetition. On-repetition type limits
         the transmitted on-change message to one message at a certain
         interval.";
    }

    identity authentication-mode {
      description
        "The authentication mode for a user to connect to the NSF,
         e.g., pre-configured-key and certificate-authority";
    }
    identity pre-configured-key {
      base authentication-mode;
      description
        "The pre-configured-key is an authentication using a key
         authentication.";
    }
    identity certificate-authority {
      base authentication-mode;
      description
        "The certificate-authority (CA) is an authentication using a
         digital certificate.";
    }

    identity event {
```

```
      description
        "Base identity for I2NSF events.";
    }

    identity system-event {
      base event;
      description
        "Identity for system event";
    }

    identity system-alarm {
      base event;
      description
        "Base identity for detectable system alarm types";
    }

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

```
     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.";
     }

     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).";
     }
     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).";
     }

     identity attack-type {
       description
         "The root ID of attack-based notification
          in the notification taxonomy";
     }
     identity nsf-attack-type {
       base attack-type;
       description
         "This ID is intended to be used
          in the context of NSF event.";
     }

     identity virus-type {
       base nsf-attack-type;
       description
         "The type of virus. It can be multiple types at once.
          This attack type is associated with a detected
          system-log virus-attack.";
     }
     identity trojan {
       base virus-type;
       description
         "The virus type is a trojan. Trojan is able to disguise the
          intent of the files or programs to misleads the users.";
```

```
     }
     identity worm {
       base virus-type;
       description
         "The virus type is a worm. Worm can self-replicate and
          spread through the network automatically.";
     }
     identity macro {
       base virus-type;
       description
         "The virus type is a macro virus. Macro causes a series of
          threats automatically after the program is executed.";
     }
     identity boot-sector {
       base virus-type;
       description
         "The virus type is a boot sector virus. Boot sector is a virus
          that infects the core of the computer, affecting the startup
          process.";
     }
     identity polymorphic {
       base virus-type;
       description
         "The virus type is a polymorphic virus. Polymorphic can
          modify its version when it replicates, making it hard to
          detect.";
     }
     identity overwrite {
       base virus-type;
       description
         "The virus type is an overwrite virus. Overwrite can remove
          existing software and replace it with malicious code by
          overwriting it.";
     }
     identity resident {
       base virus-type;
       description
         "The virus-type is a resident virus. Resident saves itself in
          the computer’s memory and infects other files and software.";
     }
     identity non-resident {
       base virus-type;
       description
         "The virus-type is a non-resident virus. Non-resident attaches
          directly to an executable file and enters the device when
          executed.";
     }
     identity multipartite {
```

```
      base virus-type;
      description
        "The virus-type is a multipartite virus. Multipartite attacks
         both the boot sector and executables files of a computer.";
    }
    identity spacefiller {
      base virus-type;
      description
        "The virus-type is a spacefiller virus. Spacefiller fills empty
         spaces of a file or software with malicious code.";
    }

    identity intrusion-attack-type {
      base nsf-attack-type;
      description
        "The attack type is associated with a detected
         system-log intrusion.";
    }
    identity brute-force {
      base intrusion-attack-type;
      description
        "The intrusion type is brute-force.";
    }
    identity buffer-overflow {
      base intrusion-attack-type;
      description
        "The intrusion type is buffer-overflow.";
    }
    identity web-attack-type {
      base nsf-attack-type;
      description
        "The attack type is associated with a detected
         system-log web-attack.";
    }
    identity command-injection {
      base web-attack-type;
      description
        "The detected web attack type is command injection.";
    }
    identity xss {
      base web-attack-type;
      description
        "The detected web attack type is Cross Site Scripting (XSS).";
    }
    identity csrf {
      base web-attack-type;
      description
        "The detected web attack type is Cross Site Request Forgery.";
```

```
     }

     identity ddos-type {
       base nsf-attack-type;
       description
         "Base identity for detectable flood types";
     }
     identity syn-flood {
       base ddos-type;
       description
         "A SYN flood is detected.";
     }
     identity ack-flood {
       base ddos-type;
       description
         "An ACK flood is detected.";
     }
     identity syn-ack-flood {
       base ddos-type;
       description
         "A SYN-ACK flood is detected.";
     }
     identity fin-rst-flood {
       base ddos-type;
       description
         "A FIN-RST flood is detected.";
     }
     identity tcp-con-flood {
       base ddos-type;
       description
         "A TCP connection flood is detected.";
     }
     identity udp-flood {
       base ddos-type;
       description
         "A UDP flood is detected.";
     }
     identity icmpv4-flood {
       base ddos-type;
       description
         "An ICMPv4 flood is detected.";
     }
     identity icmpv6-flood {
       base ddos-type;
       description
         "An ICMPv6 flood is detected.";
     }
     identity http-flood {
```

```
      base ddos-type;
      description
        "An HTTP flood is detected.";
    }
    identity https-flood {
      base ddos-type;
      description
        "An HTTPS flood is detected.";
    }
    identity dns-query-flood {
      base ddos-type;
      description
        "A Domain Name System (DNS) query flood is detected.";
    }
    identity dns-reply-flood {
      base ddos-type;
      description
       "A Domain Name System (DNS) reply flood is detected.";
    }
    identity sip-flood {
      base ddos-type;
      description
        "A Session Initiation Protocol (SIP) flood is detected.";
    }
    identity tls-flood {
      base ddos-type;
      description
        "A Transport Layer Security (TLS) flood is detected";
    }
    identity ntp-amp-flood {
      base ddos-type;
      description
        "A Network Time Protocol (NTP) amplification is detected";
    }

    identity req-method {
      description
        "A set of request types in HTTP (if applicable).";
    }
    identity put {
      base req-method;
      description
        "The detected request type is PUT.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method PUT";
    }
    identity post {
```

```
      base req-method;
      description
        "The detected request type is POST.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method POST";
    }
    identity get {
      base req-method;
      description
        "The detected request type is GET.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method GET";
    }
    identity head {
      base req-method;
      description
        "The detected request type is HEAD.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method HEAD";
    }
    identity delete {
      base req-method;
      description
        "The detected request type is DELETE.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method DELETE";
    }
    identity connect {
      base req-method;
      description
        "The detected request type is CONNECT.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method CONNECT";
    }
    identity options {
      base req-method;
      description
        "The detected request type is OPTIONS.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method OPTIONS";
    }
    identity trace {
```

```
      base req-method;
      description
        "The detected request type is TRACE.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         - Request Method TRACE";
    }

    identity filter-type {
      description
        "The type of filter used to detect an attack,
         for example, a web-attack.  It can be applicable to
         more than web-attacks.";
    }
    identity allow-list {
      base filter-type;
      description
        "The applied filter type is an allow list. This filter blocks
         all connection except the specified list.";
    }
    identity deny-list {
      base filter-type;
      description
        "The applied filter type is a deny list. This filter opens all
         connection except the specified list.";
    }
    identity unknown-filter {
      base filter-type;
      description
        "The applied filter is unknown.";
    }

    identity dpi-type {
      description
        "Base identity for the type of Deep Packet Inspection (DPI).";
    }
    identity file-blocking {
      base dpi-type;
      description
        "DPI for preventing the specified file types from flowing
         in the network.";
    }
    identity data-filtering {
      base dpi-type;
      description
        "DPI for preventing sensitive information (e.g., Credit
         Card Number or Social Security Numbers) leaving a
         protected network.";
```

```
     }
     identity application-behavior-control {
       base dpi-type;
       description
         "DPI for filtering packet based on the application or
          network behavior analysis to identify malicious or
          unusual activity.";
     }

     identity protocol {
       description
         "An identity used to enable type choices in leaves
          and leaf-lists with respect to protocol metadata. This is used
          to identify the type of protocol that goes through the NSF.";
     }
     identity ip {
       base protocol;
       description
         "General IP protocol type.";
       reference
         "RFC 791: Internet Protocol
          RFC 8200: Internet Protocol, Version 6 (IPv6)";
     }
     identity ipv4 {
       base ip;
       description
         "IPv4 protocol type.";
       reference
         "RFC 791: Internet Protocol";
     }
     identity ipv6 {
       base ip;
       description
         "IPv6 protocol type.";
       reference
         "RFC 8200: Internet Protocol, Version 6 (IPv6)";
     }
     identity icmp {
       base protocol;
       description
         "Base identity for ICMPv4 and ICMPv6 condition capability";
       reference
         "RFC 792: Internet Control Message Protocol
          RFC 4443: Internet Control Message Protocol (ICMPv6)
          for the Internet Protocol Version 6 (IPv6) Specification
          - ICMPv6";
     }
     identity icmpv4 {
```

```
      base icmp;
      description
        "ICMPv4 protocol type.";
      reference
        "RFC 791: Internet Protocol
         RFC 792: Internet Control Message Protocol";
    }
    identity icmpv6 {
      base icmp;
      description
        "ICMPv6 protocol type.";
      reference
        "RFC 8200: Internet Protocol, Version 6 (IPv6)
         RFC 4443: Internet Control Message Protocol (ICMPv6)
         for the Internet Protocol Version 6 (IPv6)
         Specification";
    }
    identity transport-protocol {
      base protocol;
      description
        "Base identity for Layer 4 protocol condition capabilities,
         e.g., TCP, UDP, SCTP, DCCP, and ICMP";
    }
    identity tcp {
      base transport-protocol;
      description
        "TCP protocol type.";
      reference
        "draft-ietf-tcpm-rfc793bis-25: Transmission Control Protocol
         (TCP) Specification";
    }
    identity udp {
      base transport-protocol;
      description
        "UDP protocol type.";
      reference
        "RFC 768: User Datagram Protocol";
    }
    identity sctp {
      base transport-protocol;
      description
        "Identity for SCTP condition capabilities";
      reference
        "draft-ietf-tsvwg-rfc4960-bis-18: Stream Control Transmission
         Protocol";
    }
    identity dccp {
      base transport-protocol;
```

```
      description
        "Identity for DCCP condition capabilities";
      reference
        "RFC 4340: Datagram Congestion Control Protocol";
    }
    identity application-protocol {
      base protocol;
      description
        "Base identity for Application protocol. Note that a subset of
         application protocols (e.g., HTTP, HTTPS, FTP, POP3, and
         IMAP) are handled in this YANG module, rather than all
         the existing application protocols.";
    }
    identity http {
      base application-protocol;
      description
        "The identity for Hypertext Transfer Protocol version 1.1
         (HTTP/1.1).";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         draft-ietf-httpbis-messaging-19: HTTP/1.1";
    }
    identity https {
      base application-protocol;
      description
        "The identity for Hypertext Transfer Protocol version 1.1
         (HTTP/1.1) over TLS.";
      reference
        "draft-ietf-httpbis-semantics-19: HTTP Semantics
         draft-ietf-httpbis-messaging-19: HTTP/1.1";
    }
    identity http2 {
      base application-protocol;
      description
        "The identity for Hypertext Transfer Protocol version 2
         (HTTP/2).";
      reference
        "draft-ietf-httpbis-http2bis-07: HTTP/2";
    }
    identity https2 {
      base application-protocol;
      description
        "The identity for Hypertext Transfer Protocol version 2
         (HTTP/2) over TLS.";
      reference
        "draft-ietf-httpbis-http2bis-07: HTTP/2";
    }
    identity ftp {
```

```
      base application-protocol;
      description
        "FTP protocol type.";
      reference
        "RFC 959: File Transfer Protocol";
    }
    identity ssh {
      base application-protocol;
      description
        "SSH protocol type.";
      reference
        "RFC 6242: Using the NETCONF Protocol over Secure Shell (SSH)";
    }
    identity telnet {
      base application-protocol;
      description
        "The identity for telnet.";
      reference
        "RFC 854: Telnet Protocol";
    }
    identity smtp {
      base application-protocol;
      description
        "The identity for smtp.";
      reference
        "RFC 5321: Simple Mail Transfer Protocol (SMTP)";
    }
    identity pop3 {
      base application-protocol;
      description
        "The identity for Post Office Protocol 3 (POP3).";
      reference
        "RFC 1939: Post Office Protocol - Version 3 (POP3)";
    }
    identity pop3s {
      base application-protocol;
      description
        "The identity for Post Office Protocol 3 (POP3) over TLS";
      reference
        "RFC 1939: Post Office Protocol - Version 3 (POP3)
         RFC 2595: Using TLS with IMAP, POP3 and ACAP";
    }
    identity imap {
      base application-protocol;
      description
        "The identity for Internet Message Access Protocol (IMAP).";
      reference
        "RFC 9051: Internet Message Access Protocol (IMAP) - Version
```

```
          4rev2";
     }
     identity imaps {
       base application-protocol;
       description
         "The identity for Internet Message Access Protocol (IMAP) over
          TLS";
       reference
         "RFC 9051: Internet Message Access Protocol (IMAP) - Version
          4rev2
          RFC 2595: Using TLS with IMAP, POP3 and ACAP";
     }

     /*
      * Grouping
      */

     grouping timestamp {
       description
         "Grouping for identifying the time of the message.";
       leaf timestamp {
         type yang:date-and-time;
         description
           "Specify the time of a message being delivered.";
       }
     }

     grouping message {
       description
         "A set of common monitoring data that is needed
          as the basic information.";
       leaf message {
         type string;
         description
           "This is a freetext annotation for
            monitoring a notification's content.";
       }
       leaf language {
         type string {
           pattern '((([A-Za-z]{2,3}(-[A-Za-z]{3}(-[A-Za-z]{3})'
                 + '{0,2})?)|[A-Za-z]{4}|[A-Za-z]{5,8})(-[A-Za-z]{4})?'
                 + '(-([A-Za-z]{2}|[0-9]{3}))?(-([A-Za-z0-9]{5,8}'
                 + '|([0-9][A-Za-z0-9]{3})))*(-[0-9A-WYZa-wyz]'
                 + '(-([A-Za-z0-9]{2,8}))+)*(-[Xx](-([A-Za-z0-9]'
                 + '{1,8}))+)?|[Xx](-([A-Za-z0-9]{1,8}))+|'
                 + '(([Ee][Nn]-[Gg][Bb]-[Oo][Ee][Dd]|[Ii]-'
                 + '[Aa][Mm][Ii]|[Ii]-[Bb][Nn][Nn]|[Ii]-'
                 + '[Dd][Ee][Ff][Aa][Uu][Ll][Tt]|[Ii]-'
```

```
                + '[Ee][Nn][Oo][Cc][Hh][Ii][Aa][Nn]'
                + '|[Ii]-[Hh][Aa][Kk]|'
                + '[Ii]-[Kk][Ll][Ii][Nn][Gg][Oo][Nn]|'
                + '[Ii]-[Ll][Uu][Xx]|[Ii]-[Mm][Ii][Nn][Gg][Oo]|'
                + '[Ii]-[Nn][Aa][Vv][Aa][Jj][Oo]|[Ii]-[Pp][Ww][Nn]|'
                + '[Ii]-[Tt][Aa][Oo]|[Ii]-[Tt][Aa][Yy]|'
                + '[Ii]-[Tt][Ss][Uu]|[Ss][Gg][Nn]-[Bb][Ee]-[Ff][Rr]|'
                + '[Ss][Gg][Nn]-[Bb][Ee]-[Nn][Ll]|[Ss][Gg][Nn]-'
                + '[Cc][Hh]-[Dd][Ee])|([Aa][Rr][Tt]-'
                + '[Ll][Oo][Jj][Bb][Aa][Nn]|[Cc][Ee][Ll]-'
                + '[Gg][Aa][Uu][Ll][Ii][Ss][Hh]|'
                + '[Nn][Oo]-[Bb][Oo][Kk]|[Nn][Oo]-'
                + '[Nn][Yy][Nn]|[Zz][Hh]-[Gg][Uu][Oo][Yy][Uu]|'
                + '[Zz][Hh]-[Hh][Aa][Kk][Kk][Aa]|[Zz][Hh]-'
                + '[Mm][Ii][Nn]|[Zz][Hh]-[Mm][Ii][Nn]-'
                + '[Nn][Aa][Nn]|[Zz][Hh]-[Xx][Ii][Aa][Nn][Gg])))';
        }
        default "en-US";
        description
          "The value in this field indicates the language tag
           used for the human readable fields (i.e., '../message',
           '/i2nsf-log/i2nsf-nsf-system-access-log/output', and
           '/i2nsf-log/i2nsf-system-user-activity-log/additional-info
           /cause').
           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";
      }
    }

    grouping common-monitoring-data {
      description
        "A set of common monitoring data that is needed
        as the basic information.";

      leaf vendor-name {
        type string;
        description
          "The name of the NSF vendor. The string is unrestricted to
           identify the provider or vendor of the NSF.";
      }
      leaf device-model {
        type string;
        description
          "The model of the device, can be represented by the
           device model name or serial number. This field is used to
           identify the model of the device that provides the security
```

```
            service.";
      }
      leaf software-version {
        type string;
        description
          "The version of the software used to provide the security
           service";
      }
      leaf nsf-name {
        type union {
          type string;
          type inet:ip-address-no-zone;
        }
        mandatory true;
        description
          "The name or IP address of the NSF generating the message.
           If the given nsf-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 a different NSF to identify the NSF that
           generates the message.";
      }
    }
    grouping characteristics {
      description
        "A set of characteristics of a monitoring information.";
      leaf acquisition-method {
        type identityref {
          base acquisition-method;
        }
        description
          "The acquisition-method for characteristics";
      }
      leaf emission-type {
        when "derived-from-or-self(../acquisition-method, "
           + "'i2nsfmi:subscription')";
        type identityref {
          base emission-type;
        }
        description
          "The emission-type for characteristics. This attribute is
           used only when the acquisition-method is a 'subscription'";
      }
    }
    grouping characteristics-extended {
      description
        "An extended characteristics for the monitoring information.";
      uses characteristics;
```

```
      leaf dampening-type {
        type identityref {
          base dampening-type;
        }
        description
          "The dampening-type for characteristics";
      }
    }
    grouping i2nsf-system-alarm-type-content {
      description
        "A set of contents for alarm type notification.";
      leaf usage {
        type uint8 {
          range "0..100";
        }
        units "percent";
        description
          "Specifies the used percentage";
      }
      leaf threshold {
        type uint8 {
          range "0..100";
        }
        units "percent";
        description
          "The threshold percentage triggering the alarm or
           the event";
      }
    }
    grouping i2nsf-system-event-type-content {
      description
        "System event metadata associated with system events
         caused by user activity. This can be extended to provide
         additional information.";
      leaf user {
        type string;
        mandatory true;
        description
          "The name of a user";
      }
      leaf-list group {
        type string;
        min-elements 1;
        description
          "The group(s) to which a user belongs.";
      }
      leaf ip-address {
        type inet:ip-address-no-zone;
```

```
        mandatory true;
        description
          "The IPv4 or IPv6 address of a user that trigger the
           event.";
      }
      leaf l4-port-number {
        type inet:port-number;
        mandatory true;
        description
          "The transport layer port number used by the user.";
      }
      leaf authentication {
        type identityref {
          base authentication-mode;
        }
        description
          "The authentication-mode of a user.";
      }
    }
    grouping i2nsf-nsf-event-type-content {
      description
        "A set of common IPv4 or IPv6-related NSF event
         content elements";
      leaf dst-ip {
        type inet:ip-address-no-zone;
        description
          "The destination IPv4 or IPv6 address of the packet";
      }
      leaf dst-port {
        type inet:port-number;
        description
          "The destination port of the packet";
      }
      leaf rule-name {
        type leafref {
          path
            "/i2nsfnfi:i2nsf-security-policy"
           +"/i2nsfnfi:rules/i2nsfnfi:name";
        }
        mandatory true;
        description
          "The name of the I2NSF Policy Rule being triggered";
      }
    }
    grouping i2nsf-nsf-event-type-content-extend {
      description
        "A set of extended common IPv4 or IPv6 related NSF
         event content elements";
```

```
      leaf src-ip {
        type inet:ip-address-no-zone;
        description
          "The source IPv4 or IPv6 address of the packet or flow";
      }
      leaf src-port {
        type inet:port-number;
        description
          "The source port of the packet or flow";
      }
      uses i2nsf-nsf-event-type-content;
    }
    grouping action {
      description
        "A grouping for action.";
      leaf-list action {
        type identityref {
          base i2nsfnfi:ingress-action;
        }
        description
          "Action type: pass, drop, reject, mirror, or rate limit";
      }
    }
    grouping attack-rates {
      description
        "A set of traffic rates for monitoring attack traffic
         data";
      leaf attack-rate {
        type uint64;
        units "pps";
        description
          "The average packets per second (pps) rate of attack
           traffic";
      }
      leaf attack-throughput {
        type uint64;
        units "Bps";
        description
          "The average bytes per second (Bps) throughput of attack
           traffic";
      }
    }
    grouping traffic-rates {
      description
        "A set of traffic rates for statistics data";
      leaf discontinuity-time {
        type yang:date-and-time;
        mandatory true;
```

```
        description
          "The time on the most recent occasion at which any one or
           more of the counters suffered a discontinuity.
           If no such discontinuities have occurred since the last
           re-initialization of the local management subsystem, then
           this node contains the time the local management subsystem
           was re-initialized.";
      }
      leaf measurement-time {
        type uint32;
        units "seconds";
        description
          "The time of the measurement in seconds for the
           calculation of statistics such as traffic rate and
           throughput. The statistic attributes are measured over
           the past measurement duration before now.";
      }
      leaf total-traffic {
        type yang:counter64;
        units "packets";
        description
          "The total number of traffic packets (in and out) in the
           NSF.";
      }
      leaf in-traffic-average-rate {
        type uint64;
        units "pps";
        description
          "Inbound traffic average rate in packets per second (pps).
           The average is calculated from the start of the NSF service
           until the generation of this record.";
      }
      leaf in-traffic-peak-rate {
        type uint64;
        units "pps";
        description
          "Inbound traffic peak rate in packets per second (pps).";
      }
      leaf in-traffic-average-throughput {
        type uint64;
        units "Bps";
        description
          "Inbound traffic average throughput in bytes per second
           (Bps). The average is calculated from the start of the NSF
           service until the generation of this record.";
      }
      leaf in-traffic-peak-throughput {
        type uint64;
```

```
        units "Bps";
        description
          "Inbound traffic peak throughput in bytes per second (Bps).";
      }
      leaf out-traffic-average-rate {
        type uint64;
        units "pps";
        description
          "Outbound traffic average rate in packets per second (pps).
           The average is calculated from the start of the NSF service
           until the generation of this record.";
      }
      leaf out-traffic-peak-rate {
        type uint64;
        units "pps";
        description
         "Outbound traffic peak rate in packets per second (pps).";
      }
      leaf out-traffic-average-throughput {
        type uint64;
        units "Bps";
        description
          "Outbound traffic average throughput in bytes per second
           (Bps). The average is calculated from the start of the NSF
           service until the generation of this record.";
      }
      leaf out-traffic-peak-throughput {
        type uint64;
        units "Bps";
        description
          "Outbound traffic peak throughput in bytes per second
           (Bps).";
      }
    }
    grouping i2nsf-system-counter-type-content {
      description
        "A set of counters for an interface traffic data.";
      leaf interface-name {
        type if:interface-ref;
        description
          "Network interface name configured in an NSF";
        reference
          "RFC 8343: A YANG Data Model for Interface Management";
      }
      leaf protocol {
        type identityref {
          base protocol;
        }
```

```
        description
          "The type of network protocol for the interface counter.
           If this field is empty, then the counter includes all
           protocols (e.g., IPv4, IPv6, TCP, and UDP)";
      }
      leaf in-total-traffic-pkts {
        type yang:counter64;
        description
          "Total inbound packets";
      }
      leaf out-total-traffic-pkts {
        type yang:counter64;
        description
          "Total outbound packets";
      }
      leaf in-total-traffic-bytes {
        type uint64;
        units "bytes";
        description
          "Total inbound bytes";
      }
      leaf out-total-traffic-bytes {
        type uint64;
        units "bytes";
        description
          "Total outbound bytes";
      }
      leaf in-drop-traffic-pkts {
        type yang:counter64;
        description
          "Total inbound drop packets";
      }
      leaf out-drop-traffic-pkts {
        type yang:counter64;
        description
          "Total outbound drop packets";
      }
      leaf in-drop-traffic-bytes {
        type uint64;
        units "bytes";
        description
          "Total inbound drop bytes";
      }
      leaf out-drop-traffic-bytes {
        type uint64;
        units "bytes";
        description
          "Total outbound drop bytes";
```

```
      }
      uses traffic-rates;
    }

    grouping i2nsf-nsf-counters-type-content {
      description
        "A set of contents of a policy in an NSF.";
      leaf policy-name {
        type leafref {
          path
            "/i2nsfnfi:i2nsf-security-policy"
           +"/i2nsfnfi:name";
        }
        mandatory true;
        description
          "The name of the policy being triggered";
      }
    }

    grouping enable-notification {
      description
        "A grouping for enabling or disabling notification";
      leaf enabled {
        type boolean;
        default "true";
        description
          "Enables or Disables the notification.
           If 'true', then the notification is enabled.
           If 'false, then the notification is disabled.";
      }
    }

    grouping dampening {
      description
        "A grouping for dampening period of notification.";
      leaf dampening-period {
        type centiseconds;
        default "0";
        description
          "Specifies the minimum interval between the assembly of
           successive update records for a single receiver of a
           subscription. Whenever subscribed objects change and
           a dampening-period interval (which may be zero) has
           elapsed since the previous update record creation for
           a receiver, any subscribed objects and properties
           that have changed since the previous update record
           will have their current values marshalled and placed
           in a new update record. But if the subscribed objects change
```

```
            when the dampening-period is active, it should update the
            record without sending the notification until the dampening-
            period is finished. If multiple changes happen during the
            active dampening-period, it should update the record with
            the latest data. And at the end of the dampening-period, it
            should send the record as a notification with the latest
            updated record and restart the countdown.";
        reference
          "RFC 8641:  Subscription to YANG Notifications for
            Datastore Updates - Section 5.";
      }
    }

    /*
     * Feature Nodes
     */

    feature i2nsf-nsf-detection-ddos {
      description
        "This feature means it supports I2NSF nsf-detection-ddos
         notification";
    }
    feature i2nsf-nsf-detection-virus {
      description
        "This feature means it supports I2NSF nsf-detection-virus
         notification";
    }
    feature i2nsf-nsf-detection-intrusion {
      description
        "This feature means it supports I2NSF nsf-detection-intrusion
         notification";
    }
    feature i2nsf-nsf-detection-web-attack {
      description
        "This feature means it supports I2NSF nsf-detection-web-attack
         notification";
    }
    feature i2nsf-nsf-detection-voip-vocn {
      description
        "This feature means it supports I2NSF nsf-detection-voip-vocn
         notification";
    }
    feature i2nsf-nsf-log-dpi {
      description
        "This feature means it supports I2NSF nsf-log-dpi
         notification";
    }
```

```
     /*
      * Notification nodes
      */

     notification i2nsf-event {
       description
         "Notification for I2NSF Event. This notification provides
          general information that can be supported by most types of
          NSFs.";

       uses common-monitoring-data;
       uses message;
       uses characteristics-extended;

       choice sub-event-type {
         description
           "This choice must be augmented with cases for each allowed
            sub-event. Only 1 sub-event will be instantiated in each
            i2nsf-event message. Each case is expected to define one
            container with all the sub-event fields.";
         case i2nsf-system-detection-alarm {
           container i2nsf-system-detection-alarm {
             description
               "This notification is sent, when a system alarm
                is detected.";
             leaf alarm-category {
               type identityref {
                 base system-alarm;
               }
               description
                 "The alarm category for
                  system-detection-alarm notification";
             }
             leaf component-name {
               type string;
               description
                 "The hardware component responsible for generating
                  the message. Applicable for Hardware Failure
                  Alarm.";
             }
             leaf interface-name {
               when "derived-from-or-self(../alarm-category, "
                  + "'i2nsfmi:interface-alarm')";
               type if:interface-ref;
               description
                 "The interface name responsible for generating
                  the message. Applicable for Network Interface
                  Failure Alarm.";
```

```
              reference
                "RFC 8343: A YANG Data Model for Interface Management";
            }
            leaf interface-state {
              when "derived-from-or-self(../alarm-category, "
                 + "'i2nsfmi:interface-alarm')";
              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 severity {
              type severity;
              description
                "The severity of the alarm such as critical, high,
                 middle, and low.";
            }
            uses i2nsf-system-alarm-type-content;
          }
        }

        case i2nsf-system-detection-event {
          container i2nsf-system-detection-event {
            description
              "This notification is sent when an event in the system is
               detected, such as access violation and configuration
               change";
            leaf event-category {
              type identityref {
                base system-event;
              }
              description
                "The event category for system-detection-event";
            }
            uses i2nsf-system-event-type-content;
            list changes {
              when "derived-from-or-self(../event-category, "
                 + "'i2nsfmi:configuration-change')";
              key policy-name;
              description
                "Describes the modification that was made to the
                 configuration. This list is only applicable when the
                 event is 'configuration-change'.
                 The minimum information that must be provided is the
                 name of the policy that has been altered (added,
                 modified, or removed).
                 This list can be extended with the detailed
                 information about the specific changes made to the
                 configuration based on the implementation.";
```

```
              leaf policy-name {
                type leafref {
                  path
                    "/i2nsfnfi:i2nsf-security-policy"
                   +"/i2nsfnfi:name";
                }
                description
                  "The name of the policy configuration that has been
                   added, modified, or removed.";
              }
            }
          }
        }

        case i2nsf-traffic-flows {
          container i2nsf-traffic-flows {
            description
              "This notification is sent to inform about the traffic
               flows.";
            leaf interface-name {
              type if:interface-ref;
              description
                "The mnemonic name of the network interface";
            }
            leaf interface-type {
              type enumeration {
                enum ingress {
                  description
                    "The corresponding interface-name indicates an
                     ingress interface.";
                }
                enum egress {
                  description
                    "The corresponding interface-name indicates an
                     egress interface.";
                }
              }
              description
                "The type of a network interface such as an ingress or
                 egress interface.";
            }
            leaf src-mac {
              type yang:mac-address;
              description
                "The source MAC address of the traffic flow. This
                 information may or may not be included depending on
                 the type of traffic flow. For example, the information
                 will be useful and should be included if the traffic
```

```
                flows are traffic flows of Link Layer Discovery
                Protocol (LLDP), Address Resolution Protocol (ARP) for
                IPv4, and Neighbor Discovery Protocol (ND) for IPv6.";
              reference
                "IEEE-802.1AB: IEEE Standard for Local and metropolitan
                area networks - Station and Media Access Control
                Connectivity Discovery - Link Layer Discovery Protocol
                (LLDP)
                RFC 826: An Ethernet Address Resolution Protocol -
                Address Resolution Protocol (ARP)
                RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
                Neighbor Discovery Protocol (ND)";
            }
            leaf dst-mac {
              type yang:mac-address;
              description
                "The destination MAC address of the traffic flow. This
                information may or may not be included depending on
                the type of traffic flow. For example, the information
                will be useful and should be included if the traffic
                flows are traffic flows of Link Layer Discovery
                Protocol (LLDP), Address Resolution Protocol (ARP) for
                IPv4, and Neighbor Discovery Protocol (ND) for IPv6.";
              reference
                "IEEE-802.1AB: IEEE Standard for Local and metropolitan
                area networks - Station and Media Access Control
                Connectivity Discovery - Link Layer Discovery Protocol
                (LLDP)
                RFC 826: An Ethernet Address Resolution Protocol -
                Address Resolution Protocol (ARP)
                RFC 4861: Neighbor Discovery for IP version 6 (IPv6) -
                Neighbor Discovery Protocol (ND)";
            }
            leaf src-ip {
              type inet:ip-address-no-zone;
              description
                "The source IPv4 or IPv6 address of the traffic flow";
            }
            leaf dst-ip {
              type inet:ip-address-no-zone;
              description
                "The destination IPv4 or IPv6 address of the traffic
                flow";
            }
            leaf protocol {
              type identityref {
                base protocol;
              }
```

```
              description
                "The protocol type of a traffic flow";
            }
            leaf src-port {
              type inet:port-number;
              description
                "The transport layer source port number of the flow";
            }
            leaf dst-port {
              type inet:port-number;
              description
                "The transport layer destination port number of the
                 flow";
            }
            leaf measurement-time {
              type uint32;
              units "seconds";
              description
                "The duration of the measurement in seconds for the
                 arrival rate and arrival throughput of packets of a
                 traffic flow. These two metrics (i.e., arrival rate
                 and arrival throughput) are measured over the past
                 measurement duration before now.";
            }
            leaf arrival-rate {
              type uint64;
              units "pps";
              description
                "The arrival rate of packets of the traffic flow in
                 packets per second measured over the past
                 'measurement-time'.";
            }
            leaf arrival-throughput {
              type uint64;
              units "Bps";
              description
                "The arrival rate of packets of the traffic flow in
                 bytes per second measured over the past
                 'measurement-time'.";
            }
          }
        }

        case i2nsf-nsf-detection-session-table {
          container i2nsf-nsf-detection-session-table {
            description
              "This notification is sent, when a session table
               event is detected.";
```

```
            leaf current-session {
              type uint32;
              description
                "The number of concurrent sessions";
            }
            leaf maximum-session {
              type uint32;
              description
                "The maximum number of sessions that the session
                 table can support";
            }
            leaf threshold {
              type uint32;
              description
                "The threshold triggering the event";
            }
          }
        }
      }
    }

    notification i2nsf-log {
      description
        "Notification for I2NSF log. The notification is generated
         from the logs of the NSF.";

      uses common-monitoring-data;
      uses message;
      uses characteristics-extended;

      choice sub-logs-type {
        description
          "This choice must be augmented with cases for each allowed
           sub-logs. Only 1 sub-event will be instantiated in each
           i2nsf-logs message. Each case is expected to define one
           container with all the sub-logs fields.";
        case i2nsf-nsf-system-access-log {
          container i2nsf-nsf-system-access-log {
            description
              "The notification is sent, if there is a new system
               log entry about a system access event.";
            uses i2nsf-system-event-type-content;
            leaf operation-type {
              type operation-type;
              description
                "The operation type that the user executes";
            }
            leaf input {
```

```
              type string;
              description
                "The operation performed by a user after login. The
                 operation is a command given by a user.";
            }
            leaf output {
              type string;
              description
                "The result in text format after executing the
                 input.";
            }
          }
        }

        case i2nsf-system-res-util-log {
          container i2nsf-system-res-util-log {
            description
              "This notification is sent, if there is a new log
               entry representing resource utilization updates.";
            leaf system-status {
              type enumeration {
                enum running {
                  description
                    "The system is active and running the security
                     service.";
                }
                enum waiting {
                  description
                    "The system is active but waiting for an event to
                     provide the security service.";
                }
                enum inactive {
                  description
                    "The system is inactive and not running the
                     security service.";
                }
              }
              description
                "The current system's running status";
            }
            leaf cpu-usage {
              type uint8;
              units "percent";
              description
                "Specifies the relative percentage of CPU utilization
                 with respect to platform resources";
            }
            leaf memory-usage {
```

```
              type uint8;
              units "percent";
              description
                "Specifies the percentage of memory usage.";
            }
            list disks {
              key disk-id;
              description
                "Disk is the hardware to store information for a
                 long period, i.e., Hard Disk or Solid-State Drive.";
              leaf disk-id {
                type string;
                description
                  "The ID of the storage disk. It is a free form
                   identifier to identify the storage disk.";
              }
              leaf disk-usage {
                type uint8;
                units "percent";
                description
                  "Specifies the percentage of disk usage";
              }
              leaf disk-space-left {
                type uint8;
                units "percent";
                description
                  "Specifies the percentage of disk space left";
              }
            }
            leaf session-num {
              type uint32;
              description
                "The total number of sessions";
            }
            leaf process-num {
              type uint32;
              description
                "The total number of processes";
            }
            list interface {
              key interface-id;
              description
                "The network interface for connecting a device
                 with the network.";
              leaf interface-id {
                type string;
                description
                  "The ID of the network interface. It is a free form
```

```
                 identifier to identify the network interface.";
             }
             leaf in-traffic-rate {
               type uint64;
               units "pps";
               description
                 "The total inbound traffic rate in packets per
                  second";
             }
             leaf out-traffic-rate {
               type uint64;
               units "pps";
               description
                  "The total outbound traffic rate in packets per
                   second";
             }
             leaf in-traffic-throughput {
               type uint64;
               units "Bps";
               description
                 "The total inbound traffic throughput in bytes per
                  second";
             }
             leaf out-traffic-throughput {
               type uint64;
               units "Bps";
               description
                 "The total outbound traffic throughput in bytes per
                  second";
             }
           }
         }
       }

       case i2nsf-system-user-activity-log {
         container i2nsf-system-user-activity-log {
           description
             "This notification is sent, if there is a new user
              activity log entry.";
           uses i2nsf-system-event-type-content;
           leaf online-duration {
             type uint32;
             units "seconds";
             description
               "The duration of a user's activeness (stays in login)
                during a session.";
           }
           leaf logout-duration {
```

```
                type uint32;
                units "seconds";
                description
                  "The duration of a user's inactiveness (not in login)
                   from the last session.";
              }
              container additional-info {
                leaf type {
                  type enumeration {
                    enum successful-login {
                      description
                        "The user has succeeded in login.";
                    }
                    enum failed-login {
                      description
                        "The user has failed in login (e.g., wrong
                         password)";
                    }
                    enum logout {
                      description
                        "The user has succeeded in logout";
                    }
                    enum successful-password-changed {
                      description
                        "The password has been changed successfully";
                    }
                    enum failed-password-changed {
                      description
                        "The attempt to change password has failed";
                    }
                    enum lock {
                      description
                        "The user has been locked. A locked user cannot
                         login.";
                    }
                    enum unlock {
                      description
                        "The user has been unlocked.";
                    }
                  }
                  description
                    "User activities, e.g., Successful User Login,
                     Failed Login attempts, User Logout, Successful User
                     Password Change, Failed User Password Change, User
                     Lockout, User Unlocking, and Unknown.";
                }
                leaf cause {
                  type string;
```

```
                 description
                   "The cause of a failed user activity related to the
                    type of user activity. For example, when the 'type'
                    is failed-login, the value of this attribute can be
                    'Failed login attempt due to wrong password
                    entry'.";
               }
               description
                 "The additional information about user activity.";
             }
           }
         }
         case i2nsf-nsf-log-dpi {
           if-feature "i2nsf-nsf-log-dpi";
           container i2nsf-nsf-log-dpi {
             description
               "This notification is sent, if there is a new DPI
                event in the NSF log.";
             leaf attack-type {
               type identityref {
                 base dpi-type;
               }
               description
                 "The type of the DPI";
             }
             uses i2nsf-nsf-event-type-content-extend;
             uses action;
           }
         }
       }
     }

     notification i2nsf-nsf-event {
       description
         "Notification for I2NSF NSF Event. This notification provides
          specific information that can only be provided by an NSF
          that supports additional features (e.g., DDoS attack
          detection).";

       uses common-monitoring-data;
       uses message;
       uses characteristics-extended;

       choice sub-event-type {
         description
         "This choice must be augmented with cases for each allowed
          sub-event. Only 1 sub-event will be instantiated in each
          i2nsf-event message. Each case is expected to define one
```

```
         container with all the sub-event fields.";
        case i2nsf-nsf-detection-ddos {
          if-feature "i2nsf-nsf-detection-ddos";
          container i2nsf-nsf-detection-ddos {
            description
              "This notification is sent, when a specific flood type
               is detected.";
            leaf attack-type {
              type identityref {
                base ddos-type;
              }
              description
                "Any one of Syn flood, ACK flood, SYN-ACK flood,
                 FIN/RST flood, TCP Connection flood, UDP flood,
                 ICMP (i.e., ICMPv4 or ICMPv6) flood, HTTP flood,
                 HTTPS flood, DNS query flood, DNS reply flood, SIP
                 flood, etc.";
            }
            leaf start-time {
              type yang:date-and-time;
              mandatory true;
              description
                "The time stamp indicating when the attack started";
            }
            leaf end-time {
              type yang:date-and-time;
              description
                "The time stamp indicating when the attack ended. If
                 the attack is still undergoing when sending out the
                 notification, this field can be omitted.";
            }
            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.";
             }
             leaf rule-name {
               type leafref {
                 path
                   "/i2nsfnfi:i2nsf-security-policy"
                  +"/i2nsfnfi:rules/i2nsfnfi:name";
               }
               mandatory true;
               description
                 "The name of the I2NSF Policy Rule being triggered";
             }

             uses attack-rates;
           }
         }
         case i2nsf-nsf-detection-virus {
           if-feature "i2nsf-nsf-detection-virus";
           container i2nsf-nsf-detection-virus {
             description
               "This notification is sent, when a virus is detected.";
             uses i2nsf-nsf-event-type-content-extend;
             leaf virus-name {
               type string;
               description
                 "The name of the detected virus";
             }
             leaf virus-type {
               type identityref {
                 base 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.";
            }
          }
        }
        case i2nsf-nsf-detection-intrusion {
          if-feature "i2nsf-nsf-detection-intrusion";
          container i2nsf-nsf-detection-intrusion {
            description
              "This notification is sent, when an intrusion event
               is detected.";
            uses i2nsf-nsf-event-type-content-extend;
            leaf protocol {
              type identityref {
                base transport-protocol;
              }
              description
                "The transport protocol type for
                 nsf-detection-intrusion notification";
```

```
            }
            leaf app {
              type identityref {
                base application-protocol;
              }
              description
                "The employed application layer protocol";
            }
            leaf attack-type {
              type identityref {
                base intrusion-attack-type;
              }
              description
                "The sub attack type for intrusion attack";
            }
          }
        }
        case i2nsf-nsf-detection-web-attack {
          if-feature "i2nsf-nsf-detection-web-attack";
          container i2nsf-nsf-detection-web-attack {
            description
              "This notification is sent, when an attack event is
               detected.";
            uses i2nsf-nsf-event-type-content-extend;
            leaf attack-type {
              type identityref {
                base web-attack-type;
              }
              description
                "Concrete web attack type, e.g., SQL injection,
                 command injection, XSS, and CSRF.";
            }
            leaf req-method {
              type identityref {
                base req-method;
              }
              description
                "The HTTP method of the request, e.g., PUT or GET.";
              reference
                "draft-ietf-httpbis-semantics-19: HTTP Semantics -
                 Request Methods";
            }
            leaf req-target {
              type string;
              description
                "The HTTP Request Target. This field can be filled in
                 the format of origin-form, absolute-form,
                 authority-form, or asterisk-form";
```

```
              reference
                "draft-ietf-httpbis-messaging-19: HTTP/1.1 - Request
                 Target";
            }
            leaf-list filtering-type {
              type identityref {
                base filter-type;
              }
              description
                "URL filtering type, e.g., deny-list, allow-list,
                 and Unknown";
            }
            leaf cookies {
              type string;
              description
                "The HTTP Cookies header field of the request from
                 the user agent. Note that though cookies have many
                 historical infelicities that degrade security and
                 privacy, the Cookie and Set-Cookie header fields are
                 widely used on the Internet. Thus, 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 HTTP Host header field of the request";
              reference
                "draft-ietf-httpbis-semantics-19: HTTP Semantics - Host";
            }
            leaf response-code {
              type string;
              description
                "The HTTP Response status code";
              reference
                "IANA Website: Hypertext Transfer Protocol (HTTP)
                 Status Code Registry";
            }
          }
        }
        case i2nsf-nsf-detection-voip-vocn {
          if-feature "i2nsf-nsf-detection-voip-vocn";
          container i2nsf-nsf-detection-voip-vocn {
            description
```

```
              "This notification is sent, when a VoIP/VoCN violation
               is detected.";
            uses i2nsf-nsf-event-type-content-extend;
            leaf-list source-voice-id {
              type string;
              description
                "The detected source voice ID for VoIP and VoCN that
                 violates the security policy.";
            }
            leaf-list destination-voice-id {
              type string;
              description
                "The detected destination voice ID for VoIP and 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.";
            }
          }
        }
      }
    }
    /*
     * Data nodes
     */
    container i2nsf-counters {
      config false;
      description
        "The state data representing continuous value changes of
         information elements that occur very frequently. The value
         should be calculated from the start of the service of the
         NSF.";

      uses common-monitoring-data;
      uses timestamp;
      uses characteristics;

      list system-interface {
        key interface-name;
        description
          "Interface counters provide the visibility of traffic into
           and out of an NSF, and bandwidth usage.";
        uses i2nsf-system-counter-type-content;
      }
      list nsf-firewall {
```

```
        key policy-name;
        description
          "Firewall counters provide visibility into traffic signatures
           and bandwidth usage that correspond to the policy that is
           configured in a firewall.";
        leaf in-interface {
          type if:interface-ref;
          description
            "Inbound interface of the traffic";
        }
        leaf out-interface {
          type if:interface-ref;
          description
            "Outbound interface of the traffic";
        }
        uses i2nsf-nsf-counters-type-content;
        uses traffic-rates;
      }
      list nsf-policy-hits {
        key policy-name;
        description
          "Policy hit counters record the number of hits that traffic
           packets match a security policy. It can check if policy
           configurations are correct or not.";
        uses i2nsf-nsf-counters-type-content;
        leaf discontinuity-time {
          type yang:date-and-time;
          mandatory true;
          description
            "The time on the most recent occasion at which any one or
             more of the counters suffered a discontinuity. If no such
             discontinuities have occurred since the last
             re-initialization of the local management subsystem, then
             this node contains the time the local management subsystem
             was re-initialized.";
        }
        leaf hit-times {
          type yang:counter64;
          description
            "The number of times that the security policy matches the
             specified traffic.";
        }
      }
    }

    container i2nsf-monitoring-configuration {
      description
        "The container for configuring I2NSF monitoring.";
```

```
       container i2nsf-system-detection-alarm {
         description
           "The container for configuring I2NSF system-detection-alarm
            notification";
         uses enable-notification;
         list system-alarm {
           key alarm-type;
           description
             "Configuration for system alarm (i.e., CPU, Memory, and
              Disk Usage)";
           leaf alarm-type {
             type enumeration {
               enum cpu {
                 description
                   "To configure the CPU usage threshold to trigger the
                    cpu-alarm";
               }
               enum memory {
                 description
                   "To configure the Memory usage threshold to trigger
                    the memory-alarm";
               }
               enum disk {
                 description
                   "To configure the Disk (storage) usage threshold to
                    trigger the disk-alarm";
               }
             }
             description
               "Type of alarm to be configured. The three alarm-types
                defined here are used to configure the threshold of the
                monitoring notification. The threshold is used to
                determine when the notification should be sent.
                The other two alarms defined in the module (i.e.,
                hardware-alarm and interface-alarm) do not use any
                threshold value to create a notification. These alarms
                detect a failure or a change of state to create a
                notification.";
           }
           leaf threshold {
             type uint8 {
               range "1..100";
             }
             units "percent";
             description
               "The configuration for threshold percentage to trigger
                the alarm. The alarm will be triggered if the usage
                is exceeded the threshold.";
```

```
          }
          uses dampening;
        }
      }
      container i2nsf-system-detection-event {
        description
          "The container for configuring I2NSF system-detection-event
           notification";
        uses enable-notification;
        uses dampening;
      }
      container i2nsf-traffic-flows {
        description
          "The container for configuring I2NSF traffic-flows
           notification";
        uses dampening;
        uses enable-notification;
      }
      container i2nsf-nsf-detection-ddos {
        if-feature "i2nsf-nsf-detection-ddos";
        description
          "The container for configuring I2NSF nsf-detection-ddos
           notification";
        uses enable-notification;
        uses dampening;
      }
      container i2nsf-nsf-detection-virus {
        if-feature "i2nsf-nsf-detection-virus";
        description
          "The container for configuring I2NSF nsf-detection-virus
           notification";
        uses enable-notification;
        uses dampening;
      }
      container i2nsf-nsf-detection-session-table {
        description
          "The container for configuring I2NSF nsf-detection-session-
           table notification";
        uses enable-notification;
        uses dampening;
      }
      container i2nsf-nsf-detection-intrusion {
        if-feature "i2nsf-nsf-detection-intrusion";
        description
          "The container for configuring I2NSF nsf-detection-intrusion
           notification";
        uses enable-notification;
        uses dampening;
```

```
        }
        container i2nsf-nsf-detection-web-attack {
          if-feature "i2nsf-nsf-detection-web-attack";
          description
            "The container for configuring I2NSF nsf-detection-web-attack
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-nsf-detection-voip-vocn {
          if-feature "i2nsf-nsf-detection-voip-vocn";
          description
            "The container for configuring I2NSF nsf-detection-voip-vocn
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-nsf-system-access-log {
          description
            "The container for configuring I2NSF system-access-log
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-system-res-util-log {
          description
            "The container for configuring I2NSF system-res-util-log
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-system-user-activity-log {
          description
            "The container for configuring I2NSF system-user-activity-log
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-nsf-log-dpi {
          if-feature "i2nsf-nsf-log-dpi";
          description
            "The container for configuring I2NSF nsf-log-dpi
             notification";
          uses enable-notification;
          uses dampening;
        }
        container i2nsf-counter {
          description
```

```
          "This is used to configure the counters
           for monitoring an NSF";
        leaf period {
          type uint16;
          units "minutes";
          default 0;
          description
            "The configuration for the period interval of reporting
             the counter. If 0, then the counter period is disabled.
             If value is not 0, then the counter will be reported
             following the period value.";
        }
      }
    }
  }
  <CODE ENDS>
```

Figure 2: Data Model of Monitoring

9. I2NSF Event Stream

This section discusses the NETCONF event stream for an I2NSF NSF Monitoring subscription. The YANG module in this document supports "ietf-subscribed-notifications" YANG module [RFC8639] for subscription. The reserved event stream name for this document is "I2NSF-Monitoring". The NETCONF Server (e.g., an NSF) MUST support "I2NSF-Monitoring" event stream for an NSF data collector (e.g., Security Controller). The "I2NSF-Monitoring" event stream contains all I2NSF events described in this document.

The following XML example shows the capabilities of the event streams generated by an NSF (e.g., "NETCONF" and "I2NSF-Monitoring" event streams) for the subscription of an NSF data collector. Refer to [RFC5277] for more detailed explanation of Event Streams. The XML examples in this document follow the line breaks as per [RFC8792].

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="1"
           xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf xmlns="urn:ietf:params:xml:ns:netmod:notification">
      <streams>
        <stream>
          <name>NETCONF</name>
          <description>Default NETCONF Event Stream</description>
          <replaySupport>false</replaySupport>
        </stream>
        <stream>
          <name>I2NSF-Monitoring</name>
          <description>I2NSF Monitoring Event Stream</description>
          <replaySupport>true</replaySupport>
          <replayLogCreationTime>
            2021-04-29T09:37:39+00:00
          </replayLogCreationTime>
        </stream>
      </streams>
    </netconf>
  </data>
</rpc-reply>
```

Figure 3: Example of NETCONF Server supporting I2NSF-Monitoring Event Stream

10. XML Examples for I2NSF NSF Monitoring

This section shows XML examples of I2NSF NSF Monitoring data delivered via Monitoring Interface from an NSF. The XML examples are following the guidelines from [RFC6241] [RFC7950].

10.1. I2NSF System Detection Alarm

The following example shows an alarm triggered by Memory Usage on the server; this example XML file is delivered by an NSF to an NSF data collector:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification
 xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
  <eventTime>2021-04-29T07:43:52.181088+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 4: Example of I2NSF System Detection Alarm triggered by Memory Usage

The XML data above shows:

1. The NSF that sends the information is named "time_based_firewall".

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

3. The monitoring information is received by subscription method.

4. The monitoring information is emitted "on-change".

5. The monitoring information is dampened "on-repetition".

6. The memory usage of the NSF is 91 percent.

7. The memory threshold to trigger the alarm is 90 percent.

8. The severity level of the notification is high.

10.2. I2NSF Interface Counters

To get the I2NSF system interface counters information by query, NETCONF Client (e.g., NSF data collector) needs to initiate GET connection with NETCONF Server (e.g., NSF). The following XML file can be used to get the state data and filter the information.

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc xmlns="urn:ietf:params:xml:ns:netconf:base:1.0" message-id="1">
  <get>
    <filter
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface">
      <i2nsf-counters>
        <system-interface/>
      </i2nsf-counters>
    </filter>
  </get>
</rpc>
```

Figure 5: XML Example for NETCONF GET with System Interface Filter

The following XML file shows the reply from the NETCONF Server (e.g., NSF):

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="1"
           xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <i2nsf-counters
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface">
      <acquisition-method>query</acquisition-method>
      <system-interface>
        <discontinuity-time>
          2021-04-29T08:43:52.181088+00:00
        </discontinuity-time>
        <interface-name>ens3</interface-name>
        <in-total-traffic-bytes>549050</in-total-traffic-bytes>
        <out-total-traffic-bytes>814956</out-total-traffic-bytes>
        <in-drop-traffic-bytes>0</in-drop-traffic-bytes>
        <out-drop-traffic-bytes>5078</out-drop-traffic-bytes>
        <nsf-name>time_based_firewall</nsf-name>
      </system-interface>
      <system-interface>
        <discontinuity-time>
          2021-04-29T08:43:52.181088+00:00
        </discontinuity-time>
        <interface-name>lo</interface-name>
        <in-total-traffic-bytes>48487</in-total-traffic-bytes>
        <out-total-traffic-bytes>48487</out-total-traffic-bytes>
        <in-drop-traffic-bytes>0</in-drop-traffic-bytes>
        <out-drop-traffic-bytes>0</out-drop-traffic-bytes>
        <nsf-name>time_based_firewall</nsf-name>
      </system-interface>
    </i2nsf-counters>
  </data>
</rpc-reply>
```


Figure 6: Example of I2NSF System Interface Counters XML Information

11. 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-monitoring-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-monitoring-interface
namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-monitoring-interface
prefix: i2nsfmi
reference: RFC XXXX

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

12. Security Considerations

The YANG module described in this document defines a schema for data that is designed to be accessed via 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 by specific NETCONF or RESTCONF users to a preconfigured subset of all available NETCONF or RESTCONF protocol operations and content.

All data nodes defined in the YANG module which can be created, modified and deleted (i.e., config true, which is the default) are considered sensitive as they all could potentially impact security monitoring and mitigation activities. Write operations (e.g., edit-config) applied to these data nodes without proper protection could result in missed alarms or incorrect alarms information being returned to the NSF data collector. The following are threats that need to be considered and mitigated:

Compromised NSF with valid credentials: It can send falsified information to the NSF data collector to mislead detection or mitigation activities; and/or to hide activity. Currently, there is no in-framework mechanism to mitigate this and it is an issue for all monitoring infrastructures. It is important to keep confidential information from unauthorized persons to mitigate the possibility of compromising the NSF with this information.

Compromised NSF data collector with valid credentials: It has visibility to all collected security alarms; the entire detection and mitigation infrastructure may be suspect. It is important to keep confidential information from unauthorized persons to mitigate the possibility of compromising the NSF with this information.

Impersonating NSF: This involves a system trying to send false

information while imitating an NSF; client authentication would help the NSF data collector to identify this invalid NSF in the "push" model (NSF-to-collector), while the "pull" model (collector-to-NSF) should already be addressed with the authentication.

Impersonating NSF data collector: This is a rogue NSF data collector with which a legitimate NSF is tricked into communicating; for "push" model (NSF-to-collector), it is important to have valid credentials, without which it should not work; for "pull" model (collector-to-NSF), mutual authentication should be used to mitigate the threat.

In addition, to defend against the DDoS attack caused by a lot of NSFs sending massive notifications to the NSF data collector, the rate limiting or similar mechanisms should be considered in both an NSF and NSF data collector, whether in advance or just in the process of DDoS attack.

All of the readable data nodes in this YANG module may be considered sensitive in some network environments. These data nodes represent information consistent with the logging commonly performed in network and security operations. They may reveal the specific configuration of a network; vulnerabilities in specific systems; and the deployed security controls and their relative efficacy in detecting or mitigating an attack. To an attacker, this information could inform how to (further) compromise the network, evade detection, or confirm whether they have been observed by the network operator.

Additionally, many of the data nodes in this YANG module such as containers "i2nsf-system-user-activity-log", "i2nsf-system-detection-event", and "i2nsf-nsf-detection-voip-vocn" are privacy sensitive. They may describe specific or aggregate user activity including associating user names with specific IP addresses; or users with specific network usage. They may also describe the specific commands that were run by users and the resulting output. Any sensitive information in that command input or output will be visible to the NSF data collector and potentially other entities, and care must be taken to protect the confidentiality of such data from unauthorized parties.

13. 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, Tim Bray (IANA), Kyle Rose (TSV-ART), Dale R. Worley (Gen-ART), Melinda Shore (SecDir), Valery Smyslov (ART-ART), 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). This work was supported in part by the MSIT under the Information Technology Research Center (ITRC) support program (IITP-2021-2017-0-01633) supervised by the IITP.

14. Contributors

The following are co-authors of this document:

Chaehong Chung - Department of Electronic, Electrical and Computer Engineering, Sungkyunkwan University, 2066 Seobu-ro Jangan-gu, Suwon, Gyeonggi-do 16419, Republic of Korea, Email: darkhong@skku.edu

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

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

Dacheng Zhang - Huawei, Email: dacheng.zhang@huawei.com

Yi Wu - Aliababa Group, Email: anren.wy@alibaba-inc.com

Rakesh Kumar - Juniper Networks, 1133 Innovation Way, Sunnyvale, CA 94089, USA, Email: rkkumar@juniper.net

Anil Lohiya - Juniper Networks, Email: alohiya@juniper.net

15. References

15.1. Normative References

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

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

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

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

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

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

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

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

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

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

[RFC3877] Chisholm, S. and D. Romascanu, "Alarm Management Information Base (MIB)", RFC 3877, DOI 10.17487/RFC3877, September 2004, <https://www.rfc-editor.org/info/rfc3877>.

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

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

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

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

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

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

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

[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>.

[RFC7011] Claise, B., Ed., Trammell, B., Ed., and P. Aitken, "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information", STD 77, RFC 7011, DOI 10.17487/RFC7011, September 2013, <https://www.rfc-editor.org/info/rfc7011>.

[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>.

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

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

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

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

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

[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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[RFC8641] Clemm, A. and E. Voit, "Subscription to YANG Notifications for Datastore Updates", RFC 8641, DOI 10.17487/RFC8641, September 2019, <https://www.rfc-editor.org/info/rfc8641>.

[RFC8650] Voit, E., Rahman, R., Nilsen-Nygaard, E., Clemm, A., and A. Bierman, "Dynamic Subscription to YANG Events and Datastores over RESTCONF", RFC 8650, DOI 10.17487/RFC8650, November 2019, <https://www.rfc-editor.org/info/rfc8650>.

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

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

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

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

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

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

[I-D.ietf-i2nsf-nsf-facing-interface-dm]
Kim, J. T., Jeong, J. P., Park, 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-28, 23 May 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-facing-interface-dm-28.txt>.

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

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

15.2. Informative References

[RFC0826] Plummer, D., "An Ethernet Address Resolution Protocol: Or Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware", STD 37, RFC 826, DOI 10.17487/RFC0826, November 1982, <https://www.rfc-editor.org/info/rfc826>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)", RFC 4861, DOI 10.17487/RFC4861, September 2007, <https://www.rfc-editor.org/info/rfc4861>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <https://www.rfc-editor.org/info/rfc4949>.

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

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

[IANA-HTTP-Status-Code] Internet Assigned Numbers Authority (IANA), "Hypertext Transfer Protocol (HTTP) Status Code Registry", September 2018, <https://www.iana.org/assignments/http-status-codes/http-status-codes.xhtml>.

[IEEE-802.1AB] Institute of Electrical and Electronics Engineers, "IEEE Standard for Local and metropolitan area networks - Station and Media Access Control Connectivity Discovery", March 2016, <https://ieeexplore.ieee.org/document/7433915>.

Appendix A. Changes from draft-ietf-i2nsf-nsf-monitoring-data-model-19

The following changes are made from draft-ietf-i2nsf-nsf-monitoring-data-model-19:

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

Authors' Addresses

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

Patrick Lingga
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

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
United States of America
Phone: +1-734-604-0332
Email: shares@ndzh.com

Liang Frank Xia
Huawei
101 Software Avenue, Yuhuatai District
Nanjing
Jiangsu,
China
Email: Frank.xialiang@huawei.com

Henk Birkholz
Fraunhofer Institute for Secure Information Technology
Rheinstrasse 75
64295 Darmstadt
Germany
Email: henk.birkholz@sit.fraunhofer.de

I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 11 November 2023

S. Hyun, Ed.
Myongji University
J. Jeong, Ed.
T. Roh
S. Wi
Sungkyunkwan University
J. Park
ETRI

10 May 2023

I2NSF Registration Interface YANG Data Model for NSF Capability Registration
draft-ietf-i2nsf-registration-interface-dm-26

Abstract

This document defines a YANG data model for the Registration Interface between Security Controller and Developer's Management System (DMS) in the Interface to Network Security Functions (I2NSF) framework to register Network Security Functions (NSF) of the DMS with the Security Controller. The objective of this data model is to support NSF capability registration and query via I2NSF Registration Interface.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A number of Network Security Functions (NSF) may exist in the Interface to Network Security Functions (I2NSF) framework [RFC8329]. Since each of these NSFs likely has different security capabilities from each other, it is important to register the security capabilities of the NSFs to the Security Controller (i.e., Network Operator Management System in [RFC8329]). In addition, it is required to search NSFs of some required security capabilities on demand. As an example, if additional security capabilities are required to serve some security service request(s) from an I2NSF User, the Security Controller should be able to request the Developer's Management System (DMS) for NSFs that have the required security capabilities.

As the main focus of the YANG module defined in [I-D.ietf-i2nsf-capability-data-model] is to define the security capabilities of an NSF, it lacks in some information (e.g., network access information to an NSF) needed by the Security Controller. This information can be provided by the DMS as it is the vendor system that provides and deploys the NSFs. Hence, this document provides the I2NSF Registration Interface to let the DMS register the capabilities and network access information of its NSFs with the Security Controller.

This document describes an information model (see Section 4) and a YANG [RFC7950] data model (see Section 5), which is extended from the I2NSF Capability YANG data model [I-D.ietf-i2nsf-capability-data-model], for the I2NSF Registration Interface [RFC8329] between the Security Controller and the DMS to support NSF capability registration and query via the registration interface. It also describes the operations that can be performed by the Security Controller and the DMS via the Registration Interface using the defined model. Note that in either NETCONF [RFC6241] or RESTCONF [RFC8040] parlance through the I2NSF Registration Interface, the Security Controller is the client, and the DMS is the server because the Security Controller and DMS run the client and server for either NETCONF or RESTCONF, respectively.

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 following terms defined in [RFC3444], [RFC8329] and [I-D.ietf-i2nsf-capability-data-model].

* Network Security Function (NSF): A function that is responsible for a specific treatment of received packets. A Network Security Function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). Sample Network Security Service Functions are as follows: Firewall, Intrusion Prevention/ Detection System (IPS/IDS), Deep Packet Inspection (DPI), Application Visibility and Control (AVC), network virus and malware scanning, sandbox, Data Loss Prevention (DLP), Distributed Denial of Service (DDoS) mitigation and TLS proxy.

* Data Model: Data Models define managed objects at a lower level of abstraction, which include implementation- and protocol-specific details, e.g., rules that explain how to map managed objects onto lower-level protocol constructs [RFC3444].

* Information Model: Information Models are primarily useful for designers to describe the managed environment, for operators to understand the modeled objects, and for implementers as a guide to the functionality that must be described and coded in the Data Models [RFC3444].

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

3. Objectives

* Registering NSFs with the I2NSF framework: Developer’s Management System (DMS) in I2NSF framework is typically run by an NSF vendor, and uses Registration Interface to provide NSFs information (i.e., capability, specification, and access information) developed by the NSF vendor to Security Controller. Since there may be multiple vendors that provide NSFs for a target network, the I2NSF Registration Interface can be used as a standard interface for the DMSs to provide NSFs capability information to the Security Controller. For the registered NSFs, Security Controller maintains a catalog of the capabilities of those NSFs to select appropriate NSFs for the requested security services. Note that the I2NSF User and the vendor should exchange information for the discovery of Security Controller and DMS during the subscription of the security service. The I2NSF User should provide the Security Controller information (e.g., access information) to the DMS for the NSFs registration, and the vendor should provide the DMS information (e.g., access information and the types of NSFs

managed by the DMS) to the Security Controller for allowing such connections. The method of exchanging this information can be done either manually or dynamically (e.g., through the new options of I2NSF information in both DHCP [RFC2131] and DHCPv6 [RFC8415]). This actual method is out of the scope of this document.

* Updating the capabilities of registered NSFs: After an NSF is registered with Security Controller, some modifications on the capability of the NSF may be required later. In this case, DMS uses Registration Interface to deliver the update of the capability of the NSF to the Security Controller, and this update MUST be reflected on the catalog of NSFs existing in the Security Controller. That is, the DMS sends the updated NSF capability information to the Security Controller through a notification mechanism. The Security Controller updates its catalog of NSFs with the updated NSF capability information.

* Asking DMS about some required capabilities: In cases that some security capabilities are required to serve the security service request from an I2NSF User, the Security Controller searches through the registered NSFs to find ones that can provide the required capabilities. But Security Controller might fail to find any NSFs having the required capabilities among the registered NSFs. In this case, Security Controller needs to request DMS for additional NSF(s) information that can provide the required security capabilities via Registration Interface.

4. Information Model of Registration Interface

The I2NSF registration interface is used by Security Controller and Developer's Management System (DMS) in I2NSF framework. Figure 1 shows the information model of the I2NSF registration interface, which consists of two submodels: NSF capability registration and NSF capability update. Each submodel is used for the operations listed above. The remainder of this section will provide in-depth explanation of each submodel. The consideration of the design of the data model is based on the procedure and mechanism discussed in Section 8 of [I-D.ietf-i2nsf-applicability], which discusses I2NSF Framework with Network Functions Virtualization (NFV) [nfv-framework].

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      I2NSF Registration Interface Information Model                   |
|                                                                       |
|         +-+-+-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+-+-+                  |
|         | NSF Capability    |  | NSF Capability    |                  |
|         | Registration      |  | Update            |                  |
|         +-+-+-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+-+-+                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: I2NSF Registration Interface Information Model

4.1. NSF Capability Registration

This submodel is used by the DMS to register the capabilities of NSFs with the request of the Security Controller. Figure 2 shows how this submodel is constructed. The most important part in Figure 2 is the NSF capability, and this specifies the set of capabilities that the NSF to be registered can offer. The NSF Name contains a unique name of this NSF with the specified set of capabilities. The NSF name MUST be unique within the registered NSFs in the Security Controller to identify the NSF with the capability. The name can be an arbitrary string including Fully Qualified Domain Name (FQDN). To make sure each vendor does not provide a duplicated name, the name should include the vendor's detail (e.g., firewall-vendor-series_name-series_number). When registering the NSF, DMS additionally includes the network access information of the NSF which is required to enable network communications with the NSF.

The following sections will further explain the NSF capability information and the NSF access information in more detail.

```
                        +-+-+-+-+-+-+-+-+-+-+
                        | NSF Capability    |
                        | Registration      |
                        +-+-+-+-+^+-+-+-+-+-+
                                 |
           +---------------------+--------------------+
           |                     |                    |
           |                     |                    |
      +-+-+-+-+-+-+-+       +-+-+-+-+-+-+-+-+-+      +-+-+-+-+-+-+-+-+
      |   NSF     |         | NSF Capability|       | NSF Access    |
      |   Name    |         | Information   |       | Information   |
      +-+-+-+-+-+-+-+       +-+-+-+-+-+-+-+-+-+      +-+-+-+-+-+-+-+-+
```

Figure 2: NSF Capability Registration Sub-Model

4.1.1. NSF Capability Information

NSF Capability Information basically describes the security capabilities of an NSF. In Figure 3, we show capability objects of an NSF. Following the information model of NSF capabilities defined in [I-D.ietf-i2nsf-capability-data-model], we share the same I2NSF security capabilities: Directional Capabilities, Event Capabilities, Condition Capabilities, Action Capabilities, Resolution Strategy Capabilities, Default Action Capabilities. Also, NSF Capability Information additionally contains the specification of an NSF as shown in Figure 3.

```
                            +-+-+-+-+-+-+-+-+-+-+
                            | NSF Capability  |
                            |   Information   |
                            +-+-+-+-^-+-+-+-+-+
                                    |
                +-----------------------+-----------------------+
                |                                               |
                |                                               |
       +-+-+-+-+-+-+-+-+-+                            +-+-+-+-+-+-+-+-+-+
       |     I2NSF     |                            |      NSF      |
       | Capabilities  |                            | Specification |
       +-+-+-+-+-+-+-+-+-+                            +-+-+-+-+-+-+-+-+-+
                |
        +------+-------------+----------------+----------------+-------+
        |                    |                |                |       |
   +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+ |
   | Directional |  |    Event    |  |  Condition  |  |    Action   | |
   | Capabilities|  | Capabilities|  | Capabilities|  | Capabilities| |
   +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+ |
                                                                            |
                      +--------------------+--------------------+-------+
                      |                    |
                +-+-+-+-+-+-+-+-+      +-+-+-+-+-+-+-+-+
                | Resolution  |        |   Default   |
                | Strategy    |        |   Action    |
                | Capabilities|        | Capabilities|
                +-+-+-+-+-+-+-+-+      +-+-+-+-+-+-+-+-+
```

Figure 3: NSF Capability Information

4.1.1.1. NSF Specification

This information represents the specification information of an NSF. As illustrated in Figure 4, this information consists of packet processing and bandwidth capabilities.

Packet processing is the overall capability of an NSF in processing packets measured in packets per second (PPS). Bandwidth describes the information about available network amount in two cases, such as outbound and inbound. Assuming that the current throughput and packet rate statuses of each NSF are being collected through NSF monitoring [I-D.ietf-i2nsf-nsf-monitoring-data-model], these capabilities of the NSF can be used to determine whether the NSF is in congestion or not by comparing it with the current throughput of the NSF.

```
                        +-----------------+
                        |       NSF       |
                        |  Specification  |
                        +--------^--------+
                                 |
                  +--------------+-------------+
                  |                            |
                  |                            |
        +---------+---------+            +-----+-----+
        | Packet Processing |            | Bandwidth |
        +-------------------+            +-----------+
```

Figure 4: NSF Specification Overview

4.1.2. NSF Access Information

NSF Access Information contains the following that are required to communicate with an NSF through NETCONF [RFC6241] or RESTCONF [RFC8040]: an IP address (i.e., IPv4 or IPv6 address) and a port number. Note that the transport layer protocol can be any transport protocol that provides the required set of functionalities for either NETCONF or RESTCONF [RFC6241][RFC8040]. In this document, NSF Access Information is used to identify a specific NSF instance. That is, NSF Access Information is the signature (i.e., unique identifier) of an NSF instance in the overall I2NSF system.

4.2. NSF Capability Update

The deployed NSFs may require to be updated to improve the quality of the security service. The Security Controller can request update information to the DMS by subscribing to the NSF capability update notification. The DMS can send the notification of NSF capability update using the NSF capability information submodel in Section 4.1.1 for updating the capabilities of the NSF.

5. YANG Data Model of Registration Interface

5.1. YANG Tree Diagrams of Registration Interface

This section provides the YANG Tree Diagram of the I2NSF registration interface. The I2NSF Registration Interface is used by the Developer's Management System (DMS) to register NSFs and their capabilities with the Security Controller. Also, in case that the Security Controller fails to find any NSF among the registered NSFs which can provide some required capabilities, Security Controller uses the registration interface to query DMS about NSF(s) having the required capabilities. The following sections describe the YANG data models to support these operations.

Note that the YANG module in this document relies on the YANG module defined in [I-D.ietf-i2nsf-capability-data-model], hence QUIC protocol [RFC9000] and HTTP/3 are excluded in the data model. The QUIC traffic should not be treated as UDP traffic, and HTTP/3 should neither be interpreted as either HTTP/1.1 nor HTTP/2. Thus, the data model should be extended or augmented appropriately to support the handling of the QUIC protocol and HTTP/3 traffic according to the needs of the implementer.

5.1.1. NSF Capability Registration

This section describes the YANG tree diagram for the NSF capability registration and capability query.

```
NSF Capability Registration
rpcs:
  +---x nsf-capability-registration
  |  +---w input
  |  |  +---w query-nsf-capability
  |  |     +---w directional-capabilities*           identityref
  |  |     +---w event-capabilities
  |  |     |  +---w system-event-capability*   identityref
  |  |     |  +---w system-alarm-capability*   identityref
  |  |     +---w condition-capabilities
  |  |     |  +---w generic-nsf-capabilities
  |  |     |  |  +---w ethernet-capability*   identityref
  |  |     |  |  +---w ipv4-capability*       identityref
  |  |     |  |  +---w ipv6-capability*       identityref
  |  |     |  |  +---w icmpv4-capability*     identityref
  |  |     |  |  +---w icmpv6-capability*     identityref
  |  |     |  |  +---w tcp-capability*        identityref
  |  |     |  |  +---w udp-capability*        identityref
  |  |     |  |  +---w sctp-capability*       identityref
  |  |     |  |  +---w dccp-capability*       identityref
```

```
     |  |      |  +---w advanced-nsf-capabilities
     |  |      |  |  +---w anti-ddos-capability*             identityref
     |  |      |  |  +---w ips-capability*                   identityref
     |  |      |  |  +---w anti-virus-capability*            identityref
     |  |      |  |  +---w url-filtering-capability*         identityref
     |  |      |  |  +---w voip-vocn-filtering-capability*   identityref
     |  |      |  +---w context-capabilities
     |  |      |     +---w time-capabilities*                 identityref
     |  |      |     +---w application-filter-capabilities*   identityref
     |  |      |     +---w device-type-capabilities*          identityref
     |  |      |     +---w user-condition-capabilities*       identityref
     |  |      |     +---w geographic-capabilities*           identityref
     |  |      +---w action-capabilities
     |  |      |  +---w ingress-action-capability*   identityref
     |  |      |  +---w egress-action-capability*    identityref
     |  |      |  +---w log-action-capability*       identityref
     |  |      +---w resolution-strategy-capabilities*   identityref
     |  |      +---w default-action-capabilities*        identityref
     |  +--ro output
     |     +--ro nsf* [nsf-name]
     |        +--ro nsf-name                             string
     |        +--ro version?                             string
     |        +--ro directional-capabilities*            identityref
     |        +--ro event-capabilities
     |        |  +--ro system-event-capability*   identityref
     |        |  +--ro system-alarm-capability*   identityref
     |        +--ro condition-capabilities
     |        |  +--ro generic-nsf-capabilities
     |        |  |  +--ro ethernet-capability*   identityref
     |        |  |  +--ro ipv4-capability*       identityref
     |        |  |  +--ro ipv6-capability*       identityref
     |        |  |  +--ro icmpv4-capability*     identityref
     |        |  |  +--ro icmpv6-capability*     identityref
     |        |  |  +--ro tcp-capability*        identityref
     |        |  |  +--ro udp-capability*        identityref
     |        |  |  +--ro sctp-capability*       identityref
     |        |  |  +--ro dccp-capability*       identityref
     |        |  +--ro advanced-nsf-capabilities
     |        |  |  +--ro anti-ddos-capability*             identityref
     |        |  |  +--ro ips-capability*                   identityref
     |        |  |  +--ro anti-virus-capability*            identityref
     |        |  |  +--ro url-filtering-capability*         identityref
     |        |  |  +--ro voip-vocn-filtering-capability*   identityref
     |        |  +--ro context-capabilities
     |        |     +--ro time-capabilities*                 identityref
     |        |     +--ro application-filter-capabilities*   identityref
     |        |     +--ro device-type-capabilities*          identityref
     |        |     +--ro user-condition-capabilities*       identityref
```

```
     |         |     +--ro geographic-capabilities*          identityref
     |         +--ro action-capabilities
     |         |  +--ro ingress-action-capability*   identityref
     |         |  +--ro egress-action-capability*    identityref
     |         |  +--ro log-action-capability*       identityref
     |         +--ro resolution-strategy-capabilities*   identityref
     |         +--ro default-action-capabilities*        identityref
     |         +--ro nsf-specification
     |         |  +--ro packet-processing?   uint64
     |         |  +--ro bandwidth
     |         |     +--ro outbound?   uint64
     |         |     +--ro inbound?    uint64
     |         +--ro nsf-access-info
     |            +--ro ip?                    inet:ip-address-no-zone
     |            +--ro port?                  inet:port-number
     |            +--ro management-protocol?   enumeration
```

Figure 5: YANG Tree Diagram of NSF Capability Registration Module

When a Security Controller requests security services to the DMS, DMS uses the I2NSF Capability YANG Data Model [I-D.ietf-i2nsf-capability-data-model] to describe what capabilities the NSFs can offer. Security Controller makes a description of the required capabilities and then queries DMS about which NSF(s) can provide these capabilities. The DMS can apply a selection mechanism to select the NSFs that cover all requested capabilities effectively. This selection mechanism is out of the scope of this document. DMS includes the access information of the NSF which is required to make a network connection with the NSF as well as the specification of the NSFs. The NSF access information consists of ip, port, and management-protocol. The field of ip can have either an IPv4 address or an IPv6 address. The port field is used to get the transport protocol port number. As I2NSF uses a YANG data model, the management protocol can be either NETCONF or RESTCONF.

The credential management for accessing the NSFs is handled by pre-negotiation with every DMS. This management is out of the scope of this document.

The DMS can also include the resource information in terms of packet processing and bandwidth capabilities of the NSF. Detailed overview of NSF specification can be seen in Section 4.1.1.1.

5.1.2. NSF Capability Update

This section describes the YANG tree diagram for the NSF capability update.

```
     NSF Capability Update
     notifications:
       +---n nsf-capability-update
          +--ro nsf* [nsf-name]
             +--ro nsf-name                            string
             +--ro version?                            string
             +--ro directional-capabilities*           identityref
             +--ro event-capabilities
             |  +--ro system-event-capability*   identityref
             |  +--ro system-alarm-capability*   identityref
             +--ro condition-capabilities
             |  +--ro generic-nsf-capabilities
             |  |  +--ro ethernet-capability*   identityref
             |  |  +--ro ipv4-capability*       identityref
             |  |  +--ro ipv6-capability*       identityref
             |  |  +--ro icmpv4-capability*     identityref
             |  |  +--ro icmpv6-capability*     identityref
             |  |  +--ro tcp-capability*        identityref
             |  |  +--ro udp-capability*        identityref
             |  |  +--ro sctp-capability*       identityref
             |  |  +--ro dccp-capability*       identityref
             |  +--ro advanced-nsf-capabilities
             |  |  +--ro anti-ddos-capability*             identityref
             |  |  +--ro ips-capability*                   identityref
             |  |  +--ro anti-virus-capability*            identityref
             |  |  +--ro url-filtering-capability*         identityref
             |  |  +--ro voip-vocn-filtering-capability*   identityref
             |  +--ro context-capabilities
             |     +--ro time-capabilities*                 identityref
             |     +--ro application-filter-capabilities*   identityref
             |     +--ro device-type-capabilities*          identityref
             |     +--ro user-condition-capabilities*       identityref
             |     +--ro geographic-capabilities*           identityref
             +--ro action-capabilities
             |  +--ro ingress-action-capability*   identityref
             |  +--ro egress-action-capability*    identityref
             |  +--ro log-action-capability*       identityref
             +--ro resolution-strategy-capabilities*   identityref
             +--ro default-action-capabilities*        identityref
             +--ro nsf-specification
             |  +--ro packet-processing?   uint64
             |  +--ro bandwidth
             |     +--ro outbound?   uint64
             |     +--ro inbound?    uint64
             +--ro nsf-access-info
                +--ro ip?                    inet:ip-address-no-zone
                +--ro port?                  inet:port-number
                +--ro management-protocol?   enumeration
```

Figure 6: YANG Tree Diagram of NSF Capability Update Module

This YANG data model is used to update the registered NSFs. The update operation started by the Security Controller subscribing to the notification of NSFs capability updates. See [RFC8639] for the subscription mechanism. Appendix A explains the event stream for the subscription of this YANG module.

The DMS should only send the NSF(s) with updated capabilities. If an update is available, the DMS can deliver the NSF capability update notification to the Security Controller. This YANG module allows multiple NSF capability updates in a single notification. Note that the capabilities given in the update represent the full capabilities of each NSF.

5.2. YANG Module of Registration Interface

This section provides a YANG module of the data model for the registration interface between Security Controller and Developer's Management System, as defined in Section 4.

This YANG module imports from [RFC6991] and [I-D.ietf-i2nsf-capability-data-model]. It makes references to [RFC6241] [RFC8040]

```
<CODE BEGINS> file "ietf-i2nsf-registration-interface@2023-05-10.yang"
module ietf-i2nsf-registration-interface {
  yang-version 1.1;

  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface";

  prefix
    i2nsfri;

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

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

  import ietf-i2nsf-capability {
    prefix i2nsfcap;
  // RFC Ed.: replace YYYY with actual RFC number of
  // draft-ietf-i2nsf-capability-data-model and remove this note.
    reference "RFC YYYY: I2NSF Capability YANG Data Model";
```

```
     }

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

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

        Editor: Sangwon Hyun
        <mailto:shyun@mju.ac.kr>

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

     description
       "This module defines a YANG data model for I2NSF
        Registration Interface.

        The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL',
        'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED',
        'NOT RECOMMENDED', 'MAY', and 'OPTIONAL' in this
        document are to be interpreted as described in BCP 14
        (RFC 2119) (RFC 8174) when, and only when, they appear
        in all capitals, as shown here.

        Copyright (c) 2023 IETF Trust and the persons
        identified as authors of the code. All rights reserved.

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

        This version of this YANG module is part of RFC XXXX; see
        the RFC itself for full legal notices.";

     revision "2023-05-10" {
       description "Initial revision";
       reference
         "RFC XXXX: I2NSF Registration Interface YANG Data Model";
       // RFC Ed.: replace XXXX with actual RFC number and remove
       // this note
     }
```

```
     grouping nsf-specification {
       description
         "Description of the specification of an NSF.";

       leaf packet-processing {
         type uint64;
         units "pps";
         description
           "The overall packet processing capability of the NSF measured in
            packets per second (pps).";
       }

       container bandwidth {
         description
           "Network bandwidth available on an NSF
            in the unit of Bps (Bytes per second).";

         leaf outbound {
           type uint64;
           units "Bps";
           description
             "The maximum aggregate outbound network bandwidth across all
              interfaces available to the NSF in bytes per second (Bps).";
         }

         leaf inbound {
           type uint64;
           units "Bps";
           description
             "The maximum aggregate inbound network bandwidth across all
              interfaces available to the NSF in bytes per second (Bps).";
         }
       }

     }

     grouping nsf-access-info {
       description
         "Information required to access an NSF.";
       leaf ip {
         type inet:ip-address-no-zone;
         description
           "Either an IPv4 or IPv6 address of this NSF.";
       }
       leaf port {
         type inet:port-number;
         description
           "Port available on this NSF";
```

```
      }
      leaf management-protocol {
        type enumeration {
          enum NETCONF {
            description
              "Represents the management protocol NETCONF.";
            reference
              "RFC 6241: Network Configuration Protocol (NETCONF)";
          }
          enum RESTCONF {
            description
              "Represents the management protocol RESTCONF.";
            reference
              "RFC 8040: RESTCONF Protocol";
          }
        }
        description
          "The management protocol used to manage the NSF.";
      }
    }

    grouping nsf {
      description
        "The information of an NSF. It consists of the name of the NSF,
         NSF capabilities, NSF specifications, and NSF access
         information";
      leaf nsf-name {
        type string;
        description
          "The name of this registered NSF. The NSF name MUST be
           unique to identify the NSF with the capability. The name
           can be an arbitrary string including Fully Qualified
           Domain Name (FQDN).";
      }
      leaf version {
        type string;
        description
          "The NSF's current version level of the software in use.
           This string MAY indicate the specific software build date and
           target variant information.";
      }
      uses i2nsfcap:nsf-capabilities;
      container nsf-specification {
        description
          "The specification of an NSF.";
        uses nsf-specification;
      }
      container nsf-access-info {
```

```
        description
          "Network access information of this NSF.";
        uses nsf-access-info;
      }
    }

    rpc nsf-capability-registration {
      description
        "Description of the capabilities that the
         Security Controller requests to the DMS";
      input {
        container query-nsf-capability {
          description
            "The query used to request NSFs. The specified
             capabilities in this field restrict the output field.";
          uses i2nsfcap:nsf-capabilities;
          reference "RFC YYYY: I2NSF Capability YANG Data Model";
        //RFC Ed.: replace YYYY with actual RFC number of
        //draft-ietf-i2nsf-capability-data-model and remove this note.
        }
      }
      output {
        list nsf {
          key "nsf-name";
          description
            "The reply of the query to register the NSFs capabilities.
             The capabilities requested in the input field can be covered
             by multiple NSFs. This list consists of NSF(s) that cover
             every capability specified in the input field.  The
             selection method of which NSF(s) that should be listed in
             the output field depends on the implementer.  If any of
             the capabilities specified in the input field cannot be
             covered by any NSF, the reply should return an <rpc-error>
             with <error-message> of those capabilities.";
          uses nsf;
        }
      }
    }

    notification nsf-capability-update {
      description
        "This notification is sent when there are updates on the
         capabilities of one or more NSFs. The list of NSFs provided in
         this notification includes the current capabilities of each NSF.
         Note that the returned capabilities represent the full
         capabilities of each NSF.";
      list nsf {
        key "nsf-name";
```

```
        description
          "List of NSFs with capabilities updated. The returned
           capabilities are the current capabilities of the NSF.";
        uses nsf;
      }
    }
  }
  <CODE ENDS>
```

Figure 7: Registration Interface YANG Data Model

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-registration-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-registration-interface
Namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface
Prefix: i2nsfri
Reference: RFC XXXX

// RFC Ed.: replace XXXX with 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 content.

The architecture of I2NSF Framework presents a risk to the implementation of security detection and mitigation activities. The risks of externally operated NSFs are discussed in Section 4 (Threats Associated with Externally Provided NSFs) of [RFC8329]. It is important to have an authentication and authorization method between the communication of the Security Controller and the DMS. The following are threats that need to be considered and mitigated:

Compromised DMS with valid credentials: It can send falsified information to the Security Controller to mislead existing detection or mitigation devices. Currently, there is no in-framework mechanism to mitigate this, and it is an issue for such infrastructures. It is important to keep confidential information from unauthorized persons to mitigate the possibility of compromising the DMS with this information.

Impersonating DMS: This involves a system trying to send false information while imitating as a DMS; client authentication would help the Security Controller to identify this invalid DMS.

The YANG module defined in this document extends the YANG module described in [I-D.ietf-i2nsf-capability-data-model]. Hence, this document shares all the security issues that are specified in Section 9 of [I-D.ietf-i2nsf-capability-data-model].

There are a number of extended 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:

* nsf-specification: The attacker may provide incorrect information of the specification of any target NSF by modifying this.

* nsf-access-info: The attacker may provide incorrect network access information of any target NSF by modifying this.

Some of the readable extended 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:

* nsf-specification: The attacker may gather the specification information of any target NSF and misuse the information for subsequent attacks.

* nsf-access-info: The attacker may gather the network access information of any target NSF and misuse the information for subsequent attacks.

The RPC operation in this YANG module MAY be considered sensitive or vulnerable in some network environments. It is thus important to control access to this operation. The following is the operation and its sensitivity/vulnerability:

* nsf-capability-query: The attacker may exploit this RPC operation to deteriorate the availability of the DMS and/or gather the information of some interested NSFs from the DMS. Some of the product capabilities provided by a vendor may be publicly known, the DMS should provide an authentication and authorization method to make sure this node cannot be used for exploitation.

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>.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <https://www.rfc-editor.org/info/rfc2131>.

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

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <https://www.rfc-editor.org/info/rfc5277>.

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

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

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

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

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

[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>.

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

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

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[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>.

[RFC8639] Voit, E., Clemm, A., Gonzalez Prieto, A., Nilsen-Nygaard, E., and A. Tripathy, "Subscription to YANG Notifications", RFC 8639, DOI 10.17487/RFC8639, September 2019, <https://www.rfc-editor.org/info/rfc8639>.

[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>.

[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>.

8.2. Informative References

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/info/rfc3444>.

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

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

[RFC8792] Watsen, K., Auerswald, E., Farrel, A., and Q. Wu, "Handling Long Lines in Content of Internet-Drafts and RFCs", RFC 8792, DOI 10.17487/RFC8792, June 2020, <https://www.rfc-editor.org/info/rfc8792>.

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

[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>.

[nfv-framework] "Network Functions Virtualisation (NFV); Architectureal Framework", ETSI GS NFV 002 ETSI GS NFV 002 V1.1.1, October 2013.

Appendix A. I2NSF Event Stream

This section discusses the NETCONF event stream for an I2NSF Registration subscription. The YANG module in this document supports "ietf-subscribed-notifications" YANG module [RFC8639] for subscription. The reserved event stream name for this document is "I2NSF-Registration". The NETCONF Server (e.g., DMS) MUST support "I2NSF-Registration" event stream for the NETCONF Client (e.g., Security Controller). The "I2NSF-Registration" event stream contains all notifications described in this document.

The following XML example shows the capabilities of the event streams generated by the DMS (e.g., "NETCONF" and "I2NSF-Registration" event streams) for the subscription of NSF capability update. Refer to [RFC5277] for a more detailed explanation of Event Streams. The XML examples in this document follow the line breaks as per [RFC8792].

```
<?xml version="1.0" encoding="UTF-8"?>
<rpc-reply message-id="1"
           xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <data>
    <netconf xmlns="urn:ietf:params:xml:ns:netmod:notification">
      <streams>
        <stream>
          <name>NETCONF</name>
          <description>Default NETCONF Event Stream</description>
          <replaySupport>false</replaySupport>
        </stream>
        <stream>
          <name>I2NSF-Registration</name>
          <description>
            I2NSF Registration Event Stream for Capability Updates
          </description>
          <replaySupport>true</replaySupport>
          <replayLogCreationTime>
            2023-04-13T09:37:39+00:00
          </replayLogCreationTime>
        </stream>
      </streams>
    </netconf>
  </data>
</rpc-reply>
```


Figure 8: Example of NETCONF Server supporting I2NSF-Registration Event Stream

Appendix B. XML Examples of an NSF Registration with I2NSF Registration Interface Data Model

This section shows XML examples of the I2NSF Registration Interface data model for registering the capabilities in either IPv4 networks [RFC5737] or IPv6 networks [RFC3849] with Security Controller.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf-capability-registration
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <query-nsf-capability>
      <condition-capabilities>
        <generic-nsf-capabilities>
          <ipv4-capability>source-address</ipv4-capability>
          <ipv4-capability>destination-address</ipv4-capability>
        </generic-nsf-capabilities>
        <advanced-nsf-capabilities>
          <url-filtering-capability>
            user-defined
          </url-filtering-capability>
        </advanced-nsf-capabilities>
      </condition-capabilities>
      <action-capabilities>
        <ingress-action-capability>drop</ingress-action-capability>
        <egress-action-capability>drop</egress-action-capability>
      </action-capabilities>
    </query-nsf-capability>
  </nsf-capability-registration>
</rpc>
```

Figure 9: XML Examples of an NSF Query with I2NSF Registration Interface Data Model

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <nsf-name>ipv4_general_firewall</nsf-name>
    <version>1.2.0</version>
    <condition-capabilities>
      <generic-nsf-capabilities>
        <ipv4-capability>next-header</ipv4-capability>
        <ipv4-capability>source-address</ipv4-capability>
        <ipv4-capability>destination-address</ipv4-capability>
        <tcp-capability>source-port-number</tcp-capability>
        <tcp-capability>destination-port-number</tcp-capability>
      </generic-nsf-capabilities>
    </condition-capabilities>
    <action-capabilities>
      <ingress-action-capability>pass</ingress-action-capability>
      <ingress-action-capability>drop</ingress-action-capability>
      <ingress-action-capability>mirror</ingress-action-capability>
      <egress-action-capability>pass</egress-action-capability>
      <egress-action-capability>drop</egress-action-capability>
      <egress-action-capability>mirror</egress-action-capability>
```

```
    </action-capabilities>
    <nsf-specification>
      <packet-processing>14000000</packet-processing>
      <bandwidth>
        <outbound>1000000000</outbound>
        <inbound>1000000000</inbound>
      </bandwidth>
    </nsf-specification>
    <nsf-access-info>
      <ip>192.0.2.11</ip>
      <port>49152</port>
      <management-protocol>
        NETCONF
      </management-protocol>
    </nsf-access-info>
  </nsf>
  <nsf
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <nsf-name>ipv4_web_filter</nsf-name>
    <version>1.1.0</version>
    <condition-capabilities>
      <advanced-nsf-capabilities>
        <url-filtering-capability>
          user-defined
        </url-filtering-capability>
      </advanced-nsf-capabilities>
    </condition-capabilities>
    <action-capabilities>
      <ingress-action-capability>pass</ingress-action-capability>
      <ingress-action-capability>drop</ingress-action-capability>
      <ingress-action-capability>mirror</ingress-action-capability>
      <egress-action-capability>pass</egress-action-capability>
      <egress-action-capability>drop</egress-action-capability>
      <egress-action-capability>mirror</egress-action-capability>
    </action-capabilities>
    <nsf-specification>
      <packet-processing>14000000</packet-processing>
      <bandwidth>
        <outbound>1000000000</outbound>
        <inbound>1000000000</inbound>
      </bandwidth>
    </nsf-specification>
    <nsf-access-info>
      <ip>192.0.2.12</ip>
      <port>49152</port>
      <management-protocol>
        NETCONF
      </management-protocol>
```

```
    </nsf-access-info>
  </nsf>
</rpc-reply>
```

Figure 10: XML Reply for the Registration of General Firewall in an IPv4 Network and Web Filter

Figure 9 shows the query for NSF(s) that can inspect IPv4 source address, destination address, and URL. Figure 10 shows the reply for the configuration XML for registering a general firewall and a web filter in an IPv4 network [RFC5737] and their capabilities.

The general firewall registered is as follows.

1. The first instance name of the NSF is ipv4_general_firewall.

2. The version used is 1.2.0.

3. The NSF can inspect IPv4 protocol header field, source address(es), and destination address(es).

4. The NSF can inspect the port number(s) for the transport layer protocol, i.e., TCP.

5. The NSF can determine whether the packets are allowed to pass, drop, or mirror.

6. The NSF is able to process 14,000,000 packets per second.

7. The network bandwidth available on the NSF is 1 GBps for both the outbound traffic and inbound traffic.

8. The IPv4 address of the NSF is 192.0.2.11.

9. The port of the NSF is 49152 using the NETCONF protocol.

The web filter registered is as follows.

1. The first instance name of the NSF is ipv4_web_filter.

2. The version used is 1.1.0.

3. The NSF can inspect a URL matched from a user-defined URL. User can specify their own URL.

4. The NSF can determine whether the packets are allowed to pass, drop, or mirror.

5. The NSF is able to process 14,000,000 packets per second.

6. The network bandwidth available on the NSF is 1 GBps for both the outbound traffic and inbound traffic.

7. The IPv4 address of the NSF is 192.0.2.12.

8. The port of the NSF is 49152 using the NETCONF protocol.

```
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf-capability-registration
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <query-nsf-capability>
      <condition-capabilities>
        <generic-nsf-capabilities>
          <ipv6-capability>source-address</ipv6-capability>
          <ipv6-capability>destination-address</ipv6-capability>
        </generic-nsf-capabilities>
        <advanced-nsf-capabilities>
          <url-filtering-capability>
            user-defined
          </url-filtering-capability>
        </advanced-nsf-capabilities>
      </condition-capabilities>
      <action-capabilities>
        <ingress-action-capability>drop</ingress-action-capability>
        <egress-action-capability>drop</egress-action-capability>
      </action-capabilities>
    </query-nsf-capability>
  </nsf-capability-registration>
</rpc>
```

Figure 11: XML Examples of an NSF Query with I2NSF Registration Interface Data Model

```
<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <nsf-name>ipv4_general_firewall</nsf-name>
    <version>1.2.0</version>
    <condition-capabilities>
      <generic-nsf-capabilities>
        <ipv6-capability>next-header</ipv6-capability>
        <ipv6-capability>source-address</ipv6-capability>
        <ipv6-capability>destination-address</ipv6-capability>
        <tcp-capability>source-port-number</tcp-capability>
```

```
        <tcp-capability>destination-port-number</tcp-capability>
      </generic-nsf-capabilities>
    </condition-capabilities>
    <action-capabilities>
      <ingress-action-capability>pass</ingress-action-capability>
      <ingress-action-capability>drop</ingress-action-capability>
      <ingress-action-capability>mirror</ingress-action-capability>
      <egress-action-capability>pass</egress-action-capability>
      <egress-action-capability>drop</egress-action-capability>
      <egress-action-capability>mirror</egress-action-capability>
    </action-capabilities>
    <nsf-specification>
      <packet-processing>14000000</packet-processing>
      <bandwidth>
        <outbound>1000000000</outbound>
        <inbound>1000000000</inbound>
      </bandwidth>
    </nsf-specification>
    <nsf-access-info>
      <ip>2001:db8:0:1::11</ip>
      <port>49153</port>
      <management-protocol>
        NETCONF
      </management-protocol>
    </nsf-access-info>
  </nsf>
  <nsf
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
    <nsf-name>ipv6_web_filter</nsf-name>
    <version>1.1.0</version>
    <condition-capabilities>
      <advanced-nsf-capabilities>
        <url-filtering-capability>
          user-defined
        </url-filtering-capability>
      </advanced-nsf-capabilities>
    </condition-capabilities>
    <action-capabilities>
      <ingress-action-capability>pass</ingress-action-capability>
      <ingress-action-capability>drop</ingress-action-capability>
      <ingress-action-capability>mirror</ingress-action-capability>
      <egress-action-capability>pass</egress-action-capability>
      <egress-action-capability>drop</egress-action-capability>
      <egress-action-capability>mirror</egress-action-capability>
    </action-capabilities>
    <nsf-specification>
      <packet-processing>14000000</packet-processing>
      <bandwidth>
```

```
        <outbound>1000000000</outbound>
        <inbound>1000000000</inbound>
      </bandwidth>
    </nsf-specification>
    <nsf-access-info>
      <ip>2001:db8:0:1::12</ip>
      <port>49153</port>
      <management-protocol>
        NETCONF
      </management-protocol>
    </nsf-access-info>
  </nsf>
</rpc-reply>
```

Figure 12: XML Reply for the Registration of General Firewall in an IPv4 Network and Web Filter

In addition, Figure 11 and Figure 12 shows the query and reply message for the configuration XML for registering a general firewall in an IPv6 network [RFC3849] and webfilter with their capabilities.

1. The instance name of the NSF is ipv6_general_firewall.

2. The version used is 1.2.0.

3. The NSF can inspect IPv6 next header, flow direction, source address(es), and destination address(es)

4. The NSF can inspect the port number(s) and flow direction for the transport layer protocol, i.e., TCP.

5. The NSF can determine whether the packets are allowed to pass, drop, or mirror.

6. The NSF is able to process 14,000,000 packets per second.

7. The network bandwidth available on the NSF is 1 GBps for both the outbound and inbound traffics.

8. The IPv6 address of the NSF is 2001:db8:0:1::11.

9. The port of the NSF is 49153 using the NETCONF protocol.

The web filter registered is as follows.

1. The first instance name of the NSF is ipv6_web_filter.

2. The version used is 1.1.0.

3. The NSF can inspect a URL matched from a user-defined URL. User can specify their own URL.

4. The NSF can determine whether the packets are allowed to pass, drop, or mirror.

5. The NSF is able to process 14,000,000 packets per second.

6. The network bandwidth available on the NSF is 1 GBps for both the outbound traffic and inbound traffic.

7. The IPv4 address of the NSF is 2001:db8:0:1::12.

8. The port of the NSF is 49153 using the NETCONF protocol.

Appendix C. XML Examples of an NSF Capability Update with I2NSF Registration Interface Data Model

This section shows an XML example of a capability update for an NSF. In this example, the registered General Firewall for the IPv4 network shown in Figure 10 is updated. The DMS can send a notification for capability update with the following XML:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<notification xmlns="urn:ietf:params:xml:ns:netconf:notification:1.0">
 <eventTime>2023-04-14T07:43:52.181088+00:00</eventTime>
 <nsf-capability-update
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-registration-interface">
  <nsf>
    <nsf-name>ipv4_general_firewall</nsf-name>
    <version>2.0.0</version>
    <condition-capabilities>
      <generic-nsf-capabilities>
        <ipv6-capability>next-header</ipv6-capability>
        <ipv6-capability>source-address</ipv6-capability>
        <ipv6-capability>destination-address</ipv6-capability>
        <tcp-capability>source-port-number</tcp-capability>
        <tcp-capability>destination-port-number</tcp-capability>
        <udp-capability>source-port-number</udp-capability>
        <udp-capability>destination-port-number</udp-capability>
      </generic-nsf-capabilities>
    </condition-capabilities>
    <action-capabilities>
      <ingress-action-capability>pass</ingress-action-capability>
      <ingress-action-capability>drop</ingress-action-capability>
      <ingress-action-capability>mirror</ingress-action-capability>
      <egress-action-capability>pass</egress-action-capability>
      <egress-action-capability>drop</egress-action-capability>
      <egress-action-capability>mirror</egress-action-capability>
    </action-capabilities>
    <nsf-specification>
      <packet-processing>14000000</packet-processing>
      <bandwidth>
        <outbound>1000000000</outbound>
        <inbound>1000000000</inbound>
      </bandwidth>
    </nsf-specification>
    <nsf-access-info>
      <ip>2001:db8:0:1::11</ip>
      <port>49153</port>
      <management-protocol>
        NETCONF
      </management-protocol>
    </nsf-access-info>
  </nsf>
 </nsf-capability-update>
</notification>
```


Figure 13: XML example of NSF capability update notification

Figure 13 shows the XML of an NSF capability update for the NSF named ipv4_general_firewall. In this example, the NSF has been updated with a new version (i.e., 2.0.0) and extended capabilities (i.e., inspect the port number(s) for UDP packets).

Appendix D. NSF Lifecycle Management in NFV Environments

Network Functions Virtualization (called NFV) can be used to implement I2NSF framework. In NFV environments, NSFs are deployed as virtual network functions (VNFs). Security Controller can be implemented as an Element Management (EM) of the NFV architecture, and is connected with the VNF Manager (VNFM) via the Ve-Vnfm interface [nfv-framework]. Security Controller can use this interface for the purpose of the lifecycle management of NSFs. If some NSFs need to be instantiated to enforce security policies in the I2NSF framework, Security Controller could request the VNFM to instantiate them through the DMS having the Ve-Vnfm interface with the VNFM. Refer to Section 8 of [I-D.ietf-i2nsf-applicability] for the detailed description on I2NSF Framework with NFV. Or if an NSF, running as a VNF, is not used by any flows for a time period, Security Controller may request deinstantiating it through the DMS having the Ve-Vnfm interface with the VNFM for efficient resource utilization.

Appendix E. 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, Reshad Rahman (YANG doctor), and Tom Petch. We 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) (No. 2016-0-00078, Cloud Based Security Intelligence Technology Development for the Customized Security Service Provisioning). This work was supported in part by the IITP (2020-0-00395-003, Standard Development of Blockchain based Network Management Automation Technology).

Appendix F. Contributors

The following are co-authors of this document:

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

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

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

Susan Hares - Huawei, 7453 Hickory Hill, Saline, MI 48176, USA. EMail: shares@ndzh.com

Diego R. Lopez - Telefonica I+D, Jose Manuel Lara, 9, Seville, 41013, Spain. EMail: diego.r.lopez@telefonica.com

Appendix G. Changes from draft-ietf-i2nsf-registration-interface-dm-25

The following changes are made from draft-ietf-i2nsf-registration-interface-dm-25:

* This version replaces Network Management Operator System with Network Operator Management System according to [RFC8329]

Authors' Addresses

Sangwon Hyun (editor)
Department of Computer Engineering
Myongji University
116 Myongji-ro, Cheoin-gu
Yongin
Gyeonggi-do
17058
Republic of Korea
Email: shyun@mju.ac.kr

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

Taekyun Roh
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 290 7222
Email: tkroh0198@skku.edu

Sarang Wi
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 290 7222
Email: dnl9795@skku.edu

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

I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: 10 August 2024

J. Jeong, Ed.
P. Lingga
J. Yang
Sungkyunkwan University

7 February 2024

Guidelines for Security Policy Translation in Interface to Network Security Functions
draft-yang-i2nsf-security-policy-translation-16

Abstract

This document proposes the guidelines for security policy translation in Interface to Network Security Functions (I2NSF) Framework. When I2NSF User delivers a high-level security policy for a security service, Security Policy Translator in Security Controller translates it into a low-level security policy for Network Security Functions (NSFs). For this security policy translation, this document specifies the relation between a high-level security policy based on the Consumer-Facing Interface YANG data model and a low-level security policy based on the NSF-Facing Interface YANG data model. Also, it describes an architecture of a security policy translator along with an NSF database, and the process of security policy translation with the NSF database.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document proposes the guidelines for security policy translation in Interface to Network Security Functions (I2NSF) Framework [RFC8329]. First of all, this document explains the necessity of a security policy translator (shortly called policy translator) in the I2NSF framework.

The policy translator resides in Security Controller in the I2NSF framework and translates a high-level security policy to a low-level security policy for Network Security Functions (NSFs). A high-level policy is specified by I2NSF User in the I2NSF framework and is delivered to Security Controller via Consumer-Facing Interface [I-D.ietf-i2nsf-consumer-facing-interface-dm]. It is translated into a low-level policy by Policy Translator in Security Controller and is delivered to NSFs to execute the rules corresponding to the low-level policy via NSF-Facing Interface [I-D.ietf-i2nsf-nsf-facing-interface-dm].

2. Terminology

This document uses the terminology specified in [RFC8329].

3. Necessity for Security Policy Translator

Security Controller acts as a coordinator between I2NSF User and NSFs. Also, Security Controller has capability information of NSFs that are registered via Registration Interface [I-D.ietf-i2nsf-registration-interface-dm] by Developer's Management System [RFC8329]. As a coordinator, Security Controller needs to generate a low-level policy in the form of security rules intended by the high-level policy, which can be understood by the corresponding NSFs.

The high-level and low-level security policies are specified by YANG data model [RFC7950] with the delivery using either NETCONF [RFC6241] or RESTCONF [RFC8040]. The translation from a high-level security policy to the corresponding low-level security policy will be able to rapidly elevate I2NSF in real-world deployment. A rule in a high-level policy can include a broad target object, such as employees in a company for a security service (e.g., firewall and web filter). Such employees may be from human resource (HR) department, software engineering department, and advertisement department. A keyword of employee needs to be mapped to these employees from various

departments. This mapping needs to be handled by a security policy translator in a flexible way while understanding the intention of a policy specification. Let us consider the following two policies:

* Block my son's computers from malicious websites.

* Drop packets from the IP address 192.0.2.0/24 to malicious1 and malicious2.

The above two sentences are examples of policies for blocking malicious websites. Both policies are for the same operation. However, NSF cannot understand the first policy, because the policy does not have any specified information for NSF. To set up the policy at an NSF, the NSF MUST receive at least the source IP address and website address for an operation. It means that the first sentence is NOT compatible for an NSF policy. Conversely, when I2NSF Users request a security policy to the system, they never make a security policy like the second example. For generating a security policy like the second sentence, the user MUST know that the NSF needs to receive the specified information, source IP address and website address. It means that the user understands the NSF professionally, but there are not many professional users in a small size of company or at a residential area. In conclusion, the I2NSF User prefers to issue a security policy in the first sentence, but an NSF will require the same policy as the second sentence with specific information. Therefore, an advanced translation scheme of security policy is REQUIRED in I2NSF.

This document proposes an approach using Automata theory [Automata] for the policy translation, such as Deterministic Finite Automaton (DFA). Note that Automata theory is the foundation of programming language and compiler. Thus, with this approach, I2NSF User can easily specify a high-level security policy that will be enforced into the corresponding NSFs with a compatibly low-level security policy with the help of Security Policy Translator. Also, for easy management, a modularized translator structure is proposed.

4. Relation between Consumer-Facing Interface and NSF-Facing Interface YANG Data Models

The Consumer-Facing Interface (CFI) YANG data model [I-D.ietf-i2nsf-consumer-facing-interface-dm] and NSF-Facing Interface (NFI) YANG data model [I-D.ietf-i2nsf-nsf-facing-interface-dm] are two data models designed with different objectives in mind. The CFI is designed to be used by someone with little knowledge of network security can configure the NSFs by specifying the required information, their data types, and encoding schemes as a high-level policy. The NFI is designed to

provide detailed security policy configuration for the NSFs as a low-level policy that can be used by the NSFs to deploy security services. But even with the distinct objectives for the data models, the attributes between the two data models are constructed to have a relation for the purpose of automation. Thus, this section provides the information of the relationship between the attributes in the CFI and NFI YANG data model.

4.1. The CFI and NFI Top-Level YANG Trees Comparison

```
Consumer-Facing Interface (CFI):
module: ietf-i2nsf-cfi-policy
  +--rw i2nsf-cfi-policy* [name]
     +--rw name                   string
     +--rw language?              string
     +--rw resolution-strategy?   identityref
     +--rw rules* [name]
     |  ...
     +--rw endpoint-groups
     |  ...
     +--rw threat-prevention
        ...

NSF-Facing Interface (NFI):
module: ietf-i2nsf-policy-rule-for-nsf
  +--rw i2nsf-security-policy* [name]
     +--rw name                   string
     +--rw language?              string
     +--rw priority-usage?        identityref
     +--rw resolution-strategy?   identityref
     +--rw default-action?        identityref
     +--rw rules* [name]
     |  ...
     +--rw rule-group
        ...
```

Figure 1: The CFI and NFI Top-Level YANG Trees

Figure 1 shows the top-level of the CFI and NFI YANG Trees. The CFI and NFI top-level provides the basic security policy information such as name of a policy, language tag, and resolution-strategy. Both data models also provide list of rules to be executed to perform the network security services.

The differences of the top-level data models are default action and priority usage are not provided in CFI YANG data model. This is because the philosophy of CFI, i.e., To make CFI as simple as possible for the user. But this attributes can be given by the

Security Controller with a default value in the translation process. Another important distinct point is CFI YANG data model also provides endpoint groups and threat prevention to register high-level information (e.g., mapping a user to an IP address) to the database for high-level configuration that can be used to translate the high-level policy into the low-level policy.

4.2. The CFI and NFI Rule-Level YANG Trees Comparison

```
Consumer-Facing Interface (CFI):
 +--rw rules* [name]
      |  +--rw name         string
      |  +--rw priority?    uint8
      |  +--rw event
      |  |  ...
      |  +--rw condition
      |  |  ...
      |  +--rw action
      |     ...

NSF-Facing Interface (NFI):
 +--rw rules* [name]
      |  +--rw name               string
      |  +--rw description?       string
      |  +--rw priority?          uint8
      |  +--rw enable?            boolean
      |  +--rw long-connection
      |  |  +--rw enable?     boolean
      |  |  +--rw duration?   uint32
      |  +--rw event
      |  |  ...
      |  +--rw condition
      |  |  ...
      |  +--rw action
      |     ...
```

Figure 2: The CFI and NFI Rule-Level YANG Trees

Figure 2 shows the rule-level YANG trees of the CFI and NFI YANG Trees. Similarly to the top-level YANG data model, the long-connection is not provided in the CFI YANG data model to simplify the data model for the user configuration. This value can also be added using a default value in the Security Controller for the low-level security policy.

In term of similarity, the CFI and NFI YANG data model provides the basic rule information such as the unique name the priority value for the rules. Both data models utilize the Event-Condition-Action (ECA) policy rule described in Section 3.1 of the [I-D.ietf-i2nsf-capability-data-model].

4.2.1. The CFI and NFI Event YANG Data Models Comparison

```
Consumer-Facing Interface (CFI):
|  +--rw event
|  |  +--rw system-event*   identityref
|  |  +--rw system-alarm*   identityref

NSF-Facing Interface (NFI):
|  +--rw event
|  |  +--rw description?     string
|  |  +--rw system-event*   identityref
|  |  +--rw system-alarm*   identityref
```

Figure 3: The CFI and NFI Event YANG Trees

As shown in Figure 3, CFI and NFI YANG data models have the almost same structures for Event except for description in NFI. The description is optional because it contains human-readable text for the description of an event.

4.2.2. The CFI and NFI Condition YANG Data Models Comparison

```
Consumer-Facing Interface (CFI):
|  +--rw condition
|  |  +--rw firewall
|  |  |  +--rw source*                     union
|  |  |  +--rw destination*                union
|  |  |  +--rw transport-layer-protocol?   identityref
|  |  |  +--rw range-port-number
|  |  |  |  +--rw start-port-number?   inet:port-number
|  |  |  |  +--rw end-port-number?     inet:port-number
|  |  |  +--rw icmp
|  |  |     +--rw message*   identityref
|  |  +--rw ddos
|  |  |  +--rw rate-limit
|  |  |     +--rw packet-rate-threshold?   uint64
|  |  |     +--rw byte-rate-threshold?     uint64
|  |  |     +--rw flow-rate-threshold?     uint64
|  |  +--rw anti-virus
|  |  |  +--rw profile*   string
|  |  |  +--rw exception-files*   string
|  |  +--rw payload
```

```
    |  |  |  +--rw content*  -> /threat-prevention/payload-content/name
    |  |  +--rw url-category
    |  |  |  +--rw url-name?  -> /endpoint-groups/url-group/name
    |  |  +--rw voice
    |  |  |  +--rw source-id*        string
    |  |  |  +--rw destination-id*   string
    |  |  |  +--rw user-agent*       string
    |  |  +--rw context
    |  |  |  +--rw time
    |  |  |  |  +--rw start-date-time?   yang:date-and-time
    |  |  |  |  +--rw end-date-time?     yang:date-and-time
    |  |  |  |  +--rw period
    |  |  |  |  |  +--rw start-time?   time
    |  |  |  |  |  +--rw end-time?     time
    |  |  |  |  |  +--rw day*          day
    |  |  |  |  |  +--rw date*         int32
    |  |  |  |  |  +--rw month*        string
    |  |  |  |  +--rw frequency?         enumeration
    |  |  |  +--rw application
    |  |  |  |  +--rw protocol*      identityref
    |  |  |  +--rw device-type
    |  |  |  |  +--rw device*        identityref
    |  |  |  +--rw users
    |  |  |  |  +--rw user* [id]
    |  |  |  |  |  +--rw id      uint32
    |  |  |  |  |  +--rw name?   string
    |  |  |  |  +--rw group* [id]
    |  |  |  |     +--rw id      uint32
    |  |  |  |     +--rw name?   string
    |  |  +--rw geographic-location
    |  |     +--rw source
    |  |     |  +--rw country? -> /endpoint-groups/location-group/country
    |  |     |  +--rw region?  -> /endpoint-groups/location-group/region
    |  |     |  +--rw city?    -> /endpoint-groups/location-group/city
    |  |     +--rw destination
    |  |        +--rw country? -> /endpoint-groups/location-group/country
    |  |        +--rw region?  -> /endpoint-groups/location-group/region
    |  |        +--rw city?    -> /endpoint-groups/location-group/city
    |  +--rw threat-feed
    |  |     +--rw name*   -> /threat-prevention/threat-feed-list/name
```

NSF-Facing Interface:

```
      |  +--rw condition
      |  |  +--rw description?    string
      |  |  +--rw layer-2* [destination-mac-address source-mac-address
                              ethertype]
      |  |  |  +--rw description?                    string
      |  |  |  +--rw destination-mac-address         yang:mac-address
```

```
          |  |  |  +--rw destination-mac-address-mask?   yang:mac-address
          |  |  |  +--rw source-mac-address              yang:mac-address
          |  |  |  +--rw source-mac-address-mask?        yang:mac-address
          |  |  |  +--rw ethertype                       eth:ethertype
          |  |  +--rw (layer-3)?
          |  |  |  +--:(ipv4)
          |  |  |  |  +--rw ipv4
          |  |  |  |     +--rw description?                      string
          |  |  |  |     +--rw dscp?                             inet:dscp
          |  |  |  |     +--rw ecn?                              uint8
          |  |  |  |     +--rw length?                           uint16
          |  |  |  |     +--rw ttl?                              uint8
          |  |  |  |     +--rw protocol?                         uint8
          |  |  |  |     +--rw ihl?                              uint8
          |  |  |  |     +--rw flags?                            bits
          |  |  |  |     +--rw offset?                           uint16
          |  |  |  |     +--rw identification?                   uint16
          |  |  |  |     +--rw (destination-network)?
          |  |  |  |     |  +--:(destination-ipv4-network)
          |  |  |  |     |  |  +--rw destination-ipv4-network?
                                                      inet:ipv4-prefix
          |  |  |  |     |  +--:(destination-ipv4-range)
          |  |  |  |     |     +--rw destination-ipv4-range* [start end]
          |  |  |  |     |        +--rw start    inet:ipv4-address-no-zone
          |  |  |  |     |        +--rw end      inet:ipv4-address-no-zone
          |  |  |  |     +--rw (source-network)?
          |  |  |  |        +--:(source-ipv4-network)
          |  |  |  |        |  +--rw source-ipv4-network? inet:ipv4-prefix
          |  |  |  |        +--:(source-ipv4-range)
          |  |  |  |           +--rw source-ipv4-range* [start end]
          |  |  |  |              +--rw start    inet:ipv4-address-no-zone
          |  |  |  |              +--rw end      inet:ipv4-address-no-zone
          |  |  |  +--:(ipv6)
          |  |  |     +--rw ipv6
          |  |  |        +--rw description?                   string
          |  |  |        +--rw dscp?                          inet:dscp
          |  |  |        +--rw ecn?                           uint8
          |  |  |        +--rw length?                        uint16
          |  |  |        +--rw ttl?                           uint8
          |  |  |        +--rw protocol?                      uint8
          |  |  |        +--rw (destination-network)?
          |  |  |        |  +--:(destination-ipv6-network)
          |  |  |        |  |  +--rw destination-ipv6-network?
                                                      inet:ipv6-prefix
          |  |  |        |  +--:(destination-ipv6-range)
          |  |  |        |     +--rw destination-ipv6-range* [start end]
          |  |  |        |        +--rw start    inet:ipv6-address-no-zone
          |  |  |        |        +--rw end      inet:ipv6-address-no-zone
```

```
          |  |  |        +--rw (source-network)?
          |  |  |           +--:(source-ipv6-network)
          |  |  |           |  +--rw source-ipv6-network? inet:ipv6-prefix
          |  |  |           +--:(source-ipv6-range)
          |  |  |              +--rw source-ipv6-range* [start end]
          |  |  |                 +--rw start    inet:ipv6-address-no-zone
          |  |  |                 +--rw end      inet:ipv6-address-no-zone
          |  |  |        +--rw flow-label?      inet:ipv6-flow-label
          |  |  +--rw (layer-4)?
          |  |  |  +--:(tcp)
          |  |  |  |  +--rw tcp
          |  |  |  |     +--rw description?               string
          |  |  |  |     +--rw source-port-number
          |  |  |  |     |  +--rw (source-port)?
          |  |  |  |     |     +--:(range-or-operator)
          |  |  |  |     |     |  +--rw (port-range-or-operator)?
          |  |  |  |     |     |     +--:(range)
          |  |  |  |     |     |     |  +--rw lower-port  inet:port-number
          |  |  |  |     |     |     |  +--rw upper-port  inet:port-number
          |  |  |  |     |     |     +--:(operator)
          |  |  |  |     |     |        +--rw operator?     operator
          |  |  |  |     |     |        +--rw port        inet:port-number
          |  |  |  |     |     +--:(port-list)
          |  |  |  |     |        +--rw port-numbers* [start end]
          |  |  |  |     |           +--rw start    inet:port-number
          |  |  |  |     |           +--rw end      inet:port-number
          |  |  |  |     +--rw destination-port-number
          |  |  |  |     |  +--rw (destination-port)?
          |  |  |  |     |     +--:(range-or-operator)
          |  |  |  |     |     |  +--rw (port-range-or-operator)?
          |  |  |  |     |     |     +--:(range)
          |  |  |  |     |     |     |  +--rw lower-port  inet:port-number
          |  |  |  |     |     |     |  +--rw upper-port  inet:port-number
          |  |  |  |     |     |     +--:(operator)
          |  |  |  |     |     |        +--rw operator?     operator
          |  |  |  |     |     |        +--rw port        inet:port-number
          |  |  |  |     |     +--:(port-list)
          |  |  |  |     |        +--rw port-numbers* [start end]
          |  |  |  |     |           +--rw start    inet:port-number
          |  |  |  |     |           +--rw end      inet:port-number
          |  |  |  |     +--rw sequence-number?           uint32
          |  |  |  |     +--rw acknowledgement-number?    uint32
          |  |  |  |     +--rw data-offset?               uint8
          |  |  |  |     +--rw reserved?                  uint8
          |  |  |  |     +--rw flags?                     bits
          |  |  |  |     +--rw window-size?               uint16
          |  |  |  |     +--rw urgent-pointer?            uint16
          |  |  |  |     +--rw options?                   binary
```

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

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

```
          |  |  |  |     +--rw service-code*              uint32
          |  |  |  |     +--rw type*                      uint8
          |  |  |  |     +--rw data-offset?               uint8
          |  |  |  +--:(icmp)
          |  |  |     +--rw icmp
          |  |  |        +--rw description?      string
          |  |  |        +--rw version?          enumeration
          |  |  |        +--rw type?             uint8
          |  |  |        +--rw code?             uint8
          |  |  |        +--rw rest-of-header?   binary
          |  |  +--rw url-category
          |  |  |  +--rw description?    string
          |  |  |  +--rw pre-defined*    string
          |  |  |  +--rw user-defined*   string
          |  |  +--rw voice
          |  |  |  +--rw description?            string
          |  |  |  +--rw source-voice-id*        string
          |  |  |  +--rw destination-voice-id*   string
          |  |  |  +--rw user-agent*             string
          |  |  +--rw ddos
          |  |  |  +--rw description?         string
          |  |  |  +--rw alert-packet-rate?   uint32
          |  |  |  +--rw alert-flow-rate?     uint32
          |  |  |  +--rw alert-byte-rate?     uint32
          |  |  +--rw anti-virus
          |  |  |  +--rw profile*           string
          |  |  |  +--rw exception-files*   string
          |  |  +--rw payload
          |  |  |  +--rw description?   string
          |  |  |  +--rw content*       binary
          |  |  +--rw context
          |  |     +--rw description?           string
          |  |     +--rw time
          |  |     |  +--rw start-date-time?   yang:date-and-time
          |  |     |  +--rw end-date-time?     yang:date-and-time
          |  |     |  +--rw period
          |  |     |  |  +--rw start-time?   time
          |  |     |  |  +--rw end-time?     time
          |  |     |  |  +--rw day*          day
          |  |     |  |  +--rw date*         int8
          |  |     |  |  +--rw month*        string
          |  |     |  +--rw frequency?         enumeration
          |  |     +--rw application
          |  |     |  +--rw description?   string
          |  |     |  +--rw protocol*      identityref
          |  |     +--rw device-type
          |  |     |  +--rw description?   string
          |  |     |  +--rw device*        identityref
```

```
        |  |     +--rw users
        |  |     |  +--rw description?   string
        |  |     |  +--rw user* [id]
        |  |     |  |  +--rw id      uint32
        |  |     |  |  +--rw name?   string
        |  |     |  +--rw group* [id]
        |  |     |     +--rw id      uint32
        |  |     |     +--rw name?   string
        |  |     +--rw geographic-location
        |  |        +--rw description?   string
        |  |        +--rw source*        string
        |  |        +--rw destination*   string
```

Figure 4: The CFI and NFI Condition YANG Trees

Figure 4 shows CFI and NFI Condition YANG Trees. It shows a different way to manipulate the Access Control Lists (ACLs) for the CFI and NFI YANG data models. The CFI aims at an easy security policy configuration, thus only provides a simple and most often needed fields in ACls, i.e., source and destination address (IPv4 or IPv6), type of transport protocol, source and destination port numbers, type of application protocol, and ICMP type and code. While, the NFI imports from [RFC8519] to provide a detailed configuration of packet header.

Additionally, both data models provide configuration for advanced network security functions such as DDoS, Antivirus, Payload (DPI), URL Filtering, and Voice Filtering conditions. The difference is that in CFI some of the information (name, value) for configuration is saved into a database in Security Controller for easy configuration. The configuration can be done by using the key name that holds the corresponding value.

The YANG data models also has context condition that can be one to one mapped, such as time condition to define the active period of a rule or geographic location condition to filter traffic from/to a certain region that can be mapped into the source and destination IP (IPv4 or IPv6) addresses based on the database provided.

4.2.3. The CFI and NFI Action YANG Data Models Comparison

```
Consumer-Facing Interface (CFI):
    +--rw action
    |  +--rw primary-action
    |  |  +--rw action?   identityref
    |  +--rw secondary-action
    |     +--rw log-action?   identityref

NSF-Facing Interface (NFI):
  |  +--rw action
  |     +--rw description?       string
  |     +--rw packet-action
  |     |  +--rw ingress-action?   identityref
  |     |  +--rw egress-action?    identityref
  |     |  +--rw log-action?       identityref
  |     +--rw flow-action
  |     |  +--rw ingress-action?   identityref
  |     |  +--rw egress-action?    identityref
  |     |  +--rw log-action?       identityref
  |     +--rw advanced-action
  |        +--rw content-security-control*    identityref
  |        +--rw attack-mitigation-control*   identityref
```

Figure 5: The CFI and NFI Action YANG Trees

Figure 4 shows CFI and NFI Action YANG Trees. The action in CFI YANG data model is separated into primary-action and secondary-action. Primary action is the Ingress and Egress action (i.e., pass, drop, reject, rate-limit, mirror, invoke-signaling, tunnel-encapsulation, forwarding, and transformation) in the NFI YANG data model. The secondary-action is the log-action to log the rule that has been triggered by a packet/flow or log the packet/flow that triggered the rule. The NFI also can specify the action as packet or flow action depending on the capability of the NSF.

In NFI YANG data model, the advanced action is used to activate the Service Function Chaining (SFC) to apply multiple NSFs on network traffics. This does not exist in CFI as the CFI is used to provide a high-level action. The action of a certain policy in CFI may require multiple NSFs (e.g., a URL filtering with firewall) as a single NSF may not have the capability to handle the security policy. Thus, the SFC of those NSFs is handled by NFI.

5. Design of Security Policy Translator

Commonly used security policies are created as XML (Extensible Markup Language) [XML] files. A popular way to change the format of an XML file is to use an XSLT (Extensible Stylesheet Language Transformation) [XSLT] document. XSLT is an XML-based language to transform an input XML file into another output XML file. However, the use of XSLT makes it difficult to manage the security policy translator and to handle the registration of new capabilities of NSFs. With the necessity for a security policy translator, this document describes a security policy translator based on Automata theory.

5.1. Overall Structure of Security Policy Translator

```
      +-------------------------------------------------+
      |                    I2NSF User                   |
      +-------------------------+-----------------------+
                                | Consumer-Facing Interface
                                |
                   High-level Security Policy
                                |
      Security Controller       V
      +-------------------------+-------------------------------+
      |  Security Policy        |                               |
      |  Translator             V                               |
      |  +----------------------+-----------------------------+ |
      |  |                      |                             | |
      |  |                      V                             | |
      |  |                +-----+-----+          +----------+ | |
      |  |                |    Data   |          |Data Model| | |
      |  |                | Extractor |          |  Mapper  | | |
      |  |                +-----+-----+          +-----+----+ | |
      |  |  Extracted Data from |              Mapping |      | |
      |  |    High-Level Policy V                Model V      | |
      |  |                +-----+-----+           +----+---+  | |
      |  |                |    Data   |<--------->| NSF DB |  | |
      |  |                | Converter |           +--------+  | |
      |  |                +-----+-----+                       | |
      |  |                      |  Required Data for          | |
      |  |                      V  Target NSFs                | |
      |  |             +--------+---------+                   | |
      |  |             | Policy Generator |                   | |
      |  |             +--------+---------+                   | |
      |  |                      |                             | |
      |  |                      V                             | |
      |  +----------------------+-----------------------------+ |
      |                         |                               |
      |                         V                               |
      +-------------------------+-------------------------------+
                                | NSF-Facing Interface
                                |
                  Low-level Security Policy
                                |
                                V
      +-------------------------+-------------------------+
      |                      NSF(s)                       |
      +-------------------------------------------------+
```

Figure 6: The Overall Design of Security Policy Translator

Figure 6 shows the overall design for Security Policy Translator in Security Controller. There are four main components for Security Policy Translator: Data Extractor, Data Converter, Policy Generator, and Data Model Mapper.

Data Model Mapper creates a mapping model for mapping the elements between Consumer-Facing Interface and NSF-Facing Interface. Data Extractor is a DFA-based module for extracting data from a high-level policy which I2NSF User delivered via Consumer-Facing Interface. Data Converter converts the extracted data to the capabilities of target NSFs for a low-level policy. It refers to an NSF Database (DB) in order to convert an abstract subject or object into the corresponding concrete subject or object (e.g., IP address and website URL). Policy Generator generates a low-level policy which will execute the NSF capabilities from Converter.

5.2. DFA-based Data Extractor

5.2.1. Design of DFA-based Data Extractor

```
                          +----------+
                          | accepter |
                          +----------+
                               | ^
                         <tag 1>| |</tag 1>
                               v |
       +-------------------------------------------------------+
       |                       middle 1                        |
       +-------------------------------------------------------+
           | ^                  | ^                       | ^
   <tag 2>| |</tag 2>  <tag 3>| |</tag 3>  ...  <tag n>| |</tag n>
          v |                  v |                       v |

          ...                 ...                     ...

     +-------------+      +-------------+          +-------------+
     | extractor 1 |      | extractor 2 |   ...    | extractor m |
     +-------------+      +-------------+          +-------------+
          data:1               data:2                   data:m
```

Figure 7: DFA Architecture of Data Extractor

Figure 7 shows a design for Data Extractor in the security policy translator. If a high-level policy contains data along the hierarchical structure of the standard Consumer-Facing Interface YANG data model [I-D.ietf-i2nsf-consumer-facing-interface-dm], data can be easily extracted using the state transition machine, such as DFA. The extracted data can be processed and used by an NSF to understand it. Extractor can be constructed by designing a DFA with the same hierarchical structure as a YANG data model.

After constructing a DFA, Data Extractor can extract all of data in the entered high-level policy by using state transitions. Also, the DFA can easily detect the grammar errors of the high-level policy. The extracting algorithm of Data Extractor is as follows:

1. Start from the 'accepter' state.

2. Read the next tag from the high-level policy.

3. Transit to the corresponding state.

4. If the current state is in 'extractor', extract the corresponding data, and then go back to step 2.

5. If the current state is in 'middle', go back to step 2.

6. If there is no possible transition and arrived at 'accepter' state, the policy has no grammar error. Otherwise, there is a grammar error, so stop the process with failure.

5.2.2. Example Scenario for Data Extractor

```
<i2nsf-cfi-policy
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-cfi-policy">
  <name>block_web_security_policy</name>
  <rules>
    <name>block_web</name>
    <condition>
      <firewall-condition>
        <source>Son's_PC</source>
      <firewall-condition>
      <url-condition>
        <url-name>malicious_websites</url-name>
      </url-condition>
    </condition>
    <actions>
      <primary-action>
        <action>drop</action>
      </primary-action>
    </actions>
  </rules>
</i2nsf-cfi-policy>
```

Figure 8: The Example of High-level Policy

```
                          +----------+
                          | accepter |
                          +----------+
                               | ^
               <i2nsf-cfi-policy>| |</i2nsf-cfi-policy>
                              v |
      +-------------------------------------------------------+
      |                       middle 1                        |
      +-------------------------------------------------------+
          | ^                                 | ^
   <name>| |</name>                           | |
         v |                                  | |
   +-------------+                  <rules>   | | </rules>
   | extractor 1 |                            | |
   +-------------+                            | |
   block_web_security                         | |
   _policy                                   v |
      +-------------------------------------------------------+
      |                   middle 2                            |
      +-------------------------------------------------------+
          | ^                     | ^                     | ^
   <name>| |</name> <condition>| | <condition>  <actions>| |</actions>
         v |                     v |                     v |
   +-------------+  +-------------------------+   +-------------+
   | extractor 2 |  |       middle 3          |   |   middle 6  |
   +-------------+  +-------------------------+   +-------------+
      block_web       | ^                 | ^           | ^
                      | |       <url-     | |</url-     | |
                      | |        condition>| | conition> | |
                      | |                 | |           | |
            <firewall-| |</firewall-      | |   <primary-| |</primary
             condition>| | condition>      | |    action>| | action>
                     v |                 v |           v |
                +-------------+   +-------------+   +-------------+
                |   middle 4  |   |   middle 5  |   |   middle 7  |
                +-------------+   +-------------+   +-------------+
                    | ^                 | ^               | ^
            <source>| |</source>  <url-| |</url-  <action>| |</action>
                    | |            name>| | name>         | |
                   v |               v |                v |
                +-------------+   +-------------+    +-------------+
                | extractor 3 |   | extractor 4 |    | extractor 5 |
                +-------------+   +-------------+    +-------------+
                   Son's_PC      malicious_websites        drop
```

Figure 9: The Example of Data Extractor

To explain the Data Extractor process by referring to an example scenario, assume that Security Controller received a high-level policy for a web-filtering as shown in Figure 8. Then we can construct a DFA-based Data Extractor by using the design as shown in Figure 7. Figure 9 shows the architecture of Data Extractor that is based on the architecture in Figure 7 along with the input high-level policy in Figure 8. Data Extractor can automatically extract all of data in the high-level policy according to the following process:

1. Start from the 'accepter' state.

2. Read the first opening tag called '<i2nsf-cfi-policy>', and transit to the 'middle 1' state.

3. Read the second opening tag called '<name>', and transit to the 'extractor 1' state.

4. The current state is an 'extractor' state. Extract the data of 'name' field called 'block_web_security_policy'.

5. Read the second closing tag called '</name>', and go back to the 'middle 1' state.

6. Read the third opening tag called '<rules>', and transit to the 'middle 2' state.

7. Read the fourth opening tag called '<name>', and transit to the 'extractor 2' state.

8. The current state is an 'extractor' state. Extract the data of 'name' field called 'block_web'.

9. Read the fourth closing tag called '</name>', and go back to the 'middle 2' state.

10. Read the fifth opening tag called '<condition>', and transit to the 'middle 3' state.

11. Read the sixth opening tag called '<firewall-condition>', and transit to the 'middle 4' state.

12. Read the seventh opening tag called '<source>', and transit to the 'extractor 3' state.

13. The current state is an 'extractor' state. Extract the data of 'source' field called 'Son's_PC'.

14. Read the seventh closing tag called '</source>', and go back to the 'middle 4' state.

15. Read the sixth closing tag called '</firewall-condition>', and go back to the 'middle 3' state.

16. Read the eight opening tag called '<url-condition>', and transit to the 'middle 5' state.

17. Read the ninth opening tag called '<url-name>', and transit to the 'extractor 4' state.

18. The current state is an 'extractor' state. Extract the data of 'url-name' field called 'malicious_websites'.

19. Read the ninth closing tag called '</url-name>', and go back to the 'middle 5' state.

20. Read the eight closing tag called '</url-condition>', and go back to the 'middle 3' state.

21. Read the fifth closing tag called '</condition>', and go back to the 'middle 2' state.

22. Read the tenth opening tag called '<actions>', and transit to the 'middle 6' state.

23. Read the eleventh opening tag called '<primary-action>', and transit to the 'middle 7' state.

24. Read the twelfth opening tag called '<action>', and transit to the 'extractor 5' state.

25. The current state is an 'extractor' state. Extract the data of 'action' field called 'drop'.

26. Read the twelfth closing tag called '</action>', and go back to the 'middle 7' state.

27. Read the eleventh closing tag called '</primary-action>', and go back to the 'middle 6' state.

28. Read the tenth closing tag called '</actions>', and go back to the 'middle 2' state.

29. Read the third closing tag called '</rules>', and go back to the 'middle 2' state.

30. Read the first closing tag called '</i2nsf-cfi-policy>', and go back to the 'accepter' state.

31. There is no further possible transition, and the state is finally on 'accepter' state. There is no grammar error in Figure 8 so the scanning for data extraction is finished.

The above process is constructed by an extracting algorithm. After finishing all the steps of the above process, Data Extractor can extract all of data in Figure 8, 'block_web_security_policy', 'block_malicious', 'Son's_PC', 'malicious_websites', and 'drop'.

Since the translator is modularized into a DFA structure, a visual understanding is feasible. Also, the performance of Data Extractor is excellent compared to one-to-one searching of data for a particular field. In addition, the management is efficient because the DFA completely follows the hierarchy of Consumer-Facing Interface. If I2NSF User wants to modify the data model of a high-level policy, it only needs to change the connection of the relevant DFA node.

5.3. Data Converter

5.3.1. Role of Data Converter

Every NSF has its own unique capabilities. The capabilities of an NSF are registered into Security Controller by a Developer's Management System, which manages the NSF, via Registration Interface. Therefore, Security Controller already has all information about the capabilities of NSFs. This means that Security Controller can find target NSFs with only the data (e.g., subject and object for a security policy) of the high-level policy by comparing the extracted data with all capabilities of each NSF. This search process for appropriate NSFs is called by policy provisioning, and it eliminates the need for I2NSF User to specify the target NSFs explicitly in a high-level security policy.

Data Converter selects target NSFs and converts the extracted data into the capabilities of selected NSFs. If Security Controller uses this data convertor, it can provide the policy provisioning function to I2NSF User automatically. Thus, the translator design provides big benefits to the I2NSF Framework.

5.3.2. NSF Database

The NSF Database contains all the information needed to convert high-level policy data to low-level policy data. The contents of NSF Database are classified as the following two: "endpoint information" and "NSF capability information".

The first is "endpoint information". Endpoint information is necessary to convert an abstract high-level policy data such as Son’s_PC and malicious_websites to a specific low-level policy data such as 192.0.2.0/24 and malicious1, respectively. In the high-level policy, the range of endpoints for applying security policy MUST be provided abstractly. Thus, endpoint information is needed to specify the abstracted high-level policy data. Endpoint information is provided by I2NSF User as the high-level policy through Consumer-Facing Interface, and Security Controller builds NSF Database based on received information.

The second is "NSF capability information". Since capability is information that allows NSF to know what features it can support, NSF capability information is used in policy provisioning process to search the appropriate NSFs through the security policy. NSF capability information is provided by Developer’s Management System (DMS) through Registration Interface, and Security Controller builds NSF Database based on received information. In addition, if the NSF sends monitoring information such as initiating information to Security Controller through NSF-Facing Interface, Security Controller can modify NSF Database accordingly.

```
     NSF Capability Information                        Endpoint Information
     +-------------------+   has        convert   +------------------+
     |        NSF        +||---+        +-------||+     Endpoint     |
     +-------------------+     |        |         +------------------+
     | *nsf_id   (INT)   |     |        |         | *end_id (INT)    |
     | nsf_name  (STRING)|     |        |         | keyword (STRING) |
     | inbound   (INT)   |     |        |         +------------------+
     | outbound  (INT)   |     |        |
     | bandwidth (INT)   |     |        |
     | activated (BOOL)  |     |        |
     +-------------------+     |        |
               +---------------+        |         +---------------------+
              /|\                       +------||+ Mapping Information |
     +--------------------+   has       |         +---------------------+
     |     Capability     +||---+       |         | *element_id (INT)   |
     +--------------------+     |       |         | element_name(STR)   |
     | *capa_id   (INT)   |     |       |         | element_map (STR)   |
     | capa_name  (STRING)|     |       |         +---------------------+
     | capa_index (INT)   |     |       |
     +--------------------+     |       |
                             /|\     /|\
                       +-----------------------+
                       |         Field         |
                       +-----------------------+
                       | *field_id   (INT)     |
                       | field_name  (STRING)  |
                       | field_index (INT)     |
                       | mapped_data (STRING)  |
                       +-----------------------+
```

Figure 10: Entity-Relationship Diagram of NSF Database

Figure 10 shows an Entity-Relationship Diagram (ERD) of NSF Database designed to include both endpoint information received from I2NSF User and NSF capability information received from DMS. By designing the NSF database based on the ERD, all the information necessary for security policy translation can be stored, and the network system administrator can manage the NSF database efficiently.

ERD was expressed by using Crow's Foot notation. Crow's Foot notation represents a relationship between entities as a line and represents the cardinality of the relationship as a symbol at both ends of the line. Attributes prefixed with * are key values of each entity. A link with two vertical lines represents one-to-one mapping, and a bird-shaped link represents one-to-many mapping. An NSF entity stores the NSF name (nsf_name), NSF specification (inbound, outbound, bandwidth), and NSF activation (activated). A Capability entity stores the capability name (capa_name) and the

index of the capability field in a Registration Interface YANG data model (capa_index). An Endpoint entity stores the keyword of abstract data conversion from I2NSF User (keyword). A Field entity stores the field name (field_name), the index of the field index in an NSF-Facing Interface YANG data model, and converted data by referring to the Endpoint entity and a 'convert' relationship.

5.3.3. Data Conversion in Data Converter

```
      High-level                            Low-level
      Policy Data                           Policy Data
   +---------------+                    +-----------------------------+
   |  Policy Name  |                    |  Policy Name                |
   | +-----------+ | The same value     | +-------------------------+ |
   | | block_web |-|------------------->|->|block_web_security_policy| |
   | | _security | |                    | +-------------------------+ |
   | |  _policy  | |                    |                             |
   | +-----------+ |                    |                             |
   |               |                    |                             |
   |  Rule Name    | The same value     |  Rule Name                  |
   | +-----------+ |                    | +-------------------------+ |
   | | block_web |-|------------------->|->|        block_web        | |
   | +-----------+ |                    | +-------------------------+ |
   |               |                    |                             |
   |    Source     | Conversion into    |  Source IPv4 Range          |
   | +-----------+ | User's IP address  | +-------------------------+ |
   | | Son's_PC  |-|------------------->|->|       192.0.2.0/24      | |
   | +-----------+ |                    | +-------------------------+ |
   |               |                    |                             |
   |  URL Name     | Conversion into    |  URL - User Defined         |
   | +-----------+ | malicious websites | +-------------------------+ |
   | | malicious |-|------------------->|->|      [malicious1,       | |
   | | _websites | |                    |  |       malicious2]       | |
   | +-----------+ |                    | +-------------------------+ |
   |               |                    |                             |
   |    Action     | Conversion into    |   Action                    |
   | +-----------+ | NSF Capability     | +-----------+               |
   | |   drop    |-|------------------->|->|drop/reject|               |
   | +-----------+ |                    | +-----------+               |
   +---------------+                    +-----------------------------+
```

Figure 11: Example of Data Conversion

Figure 11 shows an example for describing a data conversion in Data Converter. High-level policy data MUST be converted into low-level policy data which are compatible with NSFs. If a system administrator attaches a database to Data Converter, it can convert contents by referring to the database with SQL queries. Data conversion in Figure 11 is based on the following list:

* 'Policy Name' and 'Rule Name' fields do NOT need the conversion.

* 'Source' field SHOULD be converted into an IPv4 addresses.

* 'URL Name' field SHOULD be converted into a URL list of malicious websites.

* 'Action' field SHOULD be converted into the corresponding action(s) in NSF capabilities.

5.3.4. Data Model Mapper

When translating a policy, the mapping between each element of the data models are necessary to properly convert the data. The Data Model Mapper create a mapping model between the elements in Consumer-Facing Interface YANG data model and NSF-Facing Interface YANG data model. Each element in the Consumer-Facing Interface Policy Data Model has at least one or more corresponding element in NSF-Facing Interface Data Model.

```
       Consumer-Facing Interface          NSF-Facing Interface
            YANG data model                  YANG data model
                   |                                |
                   V                                V
         +---------+-------------------------------+------+
         |         |       Data Model Mapper       |      |
         |         |                               |      |
         |         |  +-------------------------+  |      |
         |         +->+ Convert as a Tree Graph +<-+      |
         |            +------------+------------+         |
         |                         |                      |
         |                         v                      |
         |          +--------------+-------------+        |
         |          | Calculate each element     |        |
         |          | Tree Edit Distance         |        |
         |          | between the CFI and NFI    |        |
         |          +--------------+-------------+        |
         |                         |                      |
         |                         v                      |
         |            +------------+------------+         |
         |            |  Get the elements with  |         |
         |            |  smallest distance as   |         |
         |            |  the candidates         |         |
         |            +------------+------------+         |
         |                         |                      |
         +-------------------------+----------------------+
                                   |
                                   V
                     Data Model Mapping Information
```

Note
CFI: Consumer-Facing Interface
NFI: NSF-Facing Interface

Figure 12: Data Model Mapping

Figure 12 shows the automatic mapping method for I2NSF Security Policy Translator. The automatic mapping is helpful as the CFI and NFI YANG data models can be extended. The automatic mapper uses the CFI and NFI YANG data models as inputs. The process the Data Model and converts it into a Tree Graph. Tree Graph is used to proces the Data Model as a Tree instead of individual elements. Then the Data Model Mapper calculates the Tree Edit Distance between each element in Consumer-Facing Interface and each element in NSF-Facing Interface. The Tree Edit Distance can be calculated with an algorithm, e.g., Zhang-Shasha algorithm [Zhang-Shasha], with the calculation should start from the root of the tree.

The Zhang-Shasha calculates the distance by three operations:

* Insert: Inserting a node or element

* Delete: Deleting a node or element

* Change: Change the label of a node or element to another

The insert and delete operations are a simple of adding/deleting a node or element with the length of the label of the node. The change operation must be calculated between the label of the element to produce the distance. There are methods to calculate this, such as Levenshtein Distance, Cosine Similarity, or Sequence Matching. For this data model mapper, cosine similarity should be the best choice as it measures the similarity between words. The data models have similarity between words and it can helps in calculating as minimum distance as possible.

When the minimum distance is obtained, the NSF-Facing Interface element is saved as the candidates for mapping the Consumer-Facing Interface element. This information should be saved to the NSF Database for the Data Converter.

Do note that the proper mapping can be achieved because the similarity between the Consumer-Facing Interface and NSF-Facing Interface. An extension created for the Consumer-Facing Interface and NSF-Facing Interface should keep the close similarity relationship between the data models to be able to produce the mapping model information automatically.

The proper mapping between CFI YANG data model and NFI YANG data model provided in [I-D.ietf-i2nsf-consumer-facing-interface-dm] and [I-D.ietf-i2nsf-nsf-facing-interface-dm], respectively, can be seen in Appendix A.

5.3.4.1. Handling of Default Values for a Low-level Security Policy

Attributes in NFI YANG data model provide detailed configuration for NSFs to handle thorough examination for security services in a network. Some of the attributes in the NFI YANG data model can be given directly after the mapping translation from a high-level policy. But the CFI YANG data model is designed to be used easily by an I2NSF User, hence some attributes cannot be mapped directly to the attributes in the NFI YANG data model.

To accommodate such attributes that cannot be given by the direct translation from the CFI YANG data model, default values can be used. For example, the attribute "default-action" in the NFI YANG data

model cannot be configured by the CFI YANG data model. A firewall usually drops packets by default to make sure that only permitted packets are allowed to pass through to the network. So, the default value for the attribute "default-action" will be a "drop" action. This can be done in the implementation of the translation so that the attribute can be given a default value.

The default value for different NSFs can be different depending on the type of service it offers. A typical firewall may use the default-value "drop" for the "default-action" attribute, but an antivirus may use a default-value "pass" for "default-action" to make sure that only the detected viruses are blocked. Other types of firewalls may also use different default values for the "default-action". Thus, the actual default values that are given to the NSFs are out of the scope of this document.

5.3.5. Policy Provisioning

```
        Log-keeper              Low-level              Web-filter
          NSF                  Policy Data               NSF
   +-------------------+  +--------------------+  +-------------------+
   |    Policy Name    |  |     Policy Name    |  |     Policy Name   |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   | | block_web    |<-|<-|<-| block_web    |->|->|->| block_web    |  |
   | | _security    |  |  | | _security     |  |  | | _security     |  |
   | | _policy      |  |  | | _policy       |  |  | | _policy       |  |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   |                   |  |                    |  |                   |
   |    Rule Name      |  |     Rule Name      |  |     Rule Name     |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   | |  block_web   |<-|<-|<-|   block_web  |->|->|->|   block_web  |  |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   |                   |  |                    |  |                   |
   |   Source IPv4     |  |    Source IPv4     |  |    Source IPv4    |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   | | 192.0.2.0/24 |<-|<-|<-| 192.0.2.0/24 |->|->|->| 192.0.2.0/24 |  |
   | +--------------+  |  | +--------------+   |  | +--------------+  |
   |                   |  |                    |  |                   |
   |                   |  | URL - User Defined |  | URL - User Defined|
   |                   |  | +--------------+   |  | +--------------+  |
   |                   |  | | [malicious1, |->|->|->| [malicious1, |  |
   |                   |  | |  malicious2] |   |  |  |  malicious2] |  |
   |                   |  | +--------------+   |  | +--------------+  |
   |                   |  |                    |  |                   |
   |    Log Action     |  |     Log Action     |  |                   |
   | +--------------+  |  | +--------------+   |  |                   |
   | |     True     |<-|<-|<-|     True     |   |  |                   |
   | +--------------+  |  | +--------------+   |  |                   |
   +-------------------+  |                    |  |                   |
                          |       Action       |  |       Action      |
                          | +--------------+   |  | +--------------+  |
                          | |     Drop     |->|->|->|  Drop/Reject |  |
                          | +--------------+   |  | +--------------+  |
                          +--------------------+  +-------------------+
```

Figure 13: Example of Policy Provisioning

Generator searches for proper NSFs which can cover all of capabilities in the high-level policy. Generator searches for target NSFs by comparing only NSF capabilities which is registered by Vendor Management System. This process is called by "policy provisioning" because Generator finds proper NSFs by using only the policy. If target NSFs are found by using other data which is not included in a user's policy, it means that the user already knows the specific knowledge of an NSF in the I2NSF Framework. Figure 13 shows an example of policy provisioning. In this example, log-keeper NSF and web-filter NSF are selected for covering capabilities in the security policy. All of capabilities can be covered by two selected NSFs.

5.4. Policy Generator

Generator makes low-level security policies for each target NSF with the extracted data. The low-level security policy can be produced in the form of XML or JSON. Libray such as PyangBind [PyangBind] for Python can be used to parse the NFI YANG data model to produce an XML or JSON form automatically.

```
+-------------------------------------------------------------+
|                     Policy Generator                        |
|                                                             |
|  +------------+      +-----------+    +-------------+      |
|  | Low-level  | Pair | Low-Level |    |  NFI YANG   |      |
|  | Attributes |<---->| Data      |    |  Data Model |      |
|  +-----+------+      +-----+-----+    +-------+-----+      |
|        |                   |                  |            |
|        |                   |                  |            |
|        +---------+---------+                  |            |
|                  |                            |            |
|                  |                            |            |
|                  v                            v            |
|          +---------------+             +------------+      |
|          |   NFI Python  |<------------| PyangBind  |      |
|          |     Class     |             +------------+      |
|          +-------+-------+                                 |
|                  |                                          |
|                  |                                          |
|                  v                                          |
|          +---------------+                                  |
|          |   Low-level   |                                  |
|          |    Policy     |                                  |
|          | (XML or JSON) |                                  |
|          +---------------+                                  |
|                                                             |
+-------------------------------------------------------------+
```

Figure 14: Policy Generator Architecture

Figure 14 shows the architecture of the Policy Generator. First, PyangBind library generates a Python class hierarchy from an input of the NFI YANG data model. This allows low-level data instances from the Data Converter (Section 5.3) to be inserted into the NFI Python Class. To get the appropriate attributes, the low-level data is paired with the attributes received from the Data Model Mapper (Section 5.3.4). The filled entry can then be encoded into an XML or JSON form automatically by PyangBind.

Figure 15 shows an XML example of a low-level policy generated by the translator.

```
<i2nsf-security-policy>
  <name>block_web_security_policy</name>
  <rules>
    <name>block_web</name>
    <condition>
      <ipv4>
        <ipv4>
          <source-ipv4-network>192.0.2.0/24</source-ipv4-network>
        </ipv4>
      </ipv4>
      <url-category>
        <user-defined>malicious1</user-defined>
        <user-defined>malicious2</user-defined>
      </url-category>
    </condition>
    <action>
      <packet-action>drop</packet-action>
    </action>
  </rules>
</i2nsf-security-policy>
```

Figure 15: Example of Low-Level Policy

6. Implementation Considerations

The implementation considerations in this document include the following three: "data model auto-adaptation", "data conversion", and "policy provisioning".

6.1. Data Model Auto-adaptation

Security Controller which acts as an intermediary entity MUST process the data according to the data model of the connected interfaces. However, the data model can be changed flexibly depending on the situation, and Security Controller may adapt to the change of the data model. Therefore, Security Controller can be implemented for convenience so that the security policy translator can easily adapt to the change of the data model.

The translator constructs and uses the DFA to adapt to the Consumer-Facing Interface Data Model. The DFA starts from the root node of the YANG tree and expands operations by changing the state according to the input. Based on the YANG data model, a container node is defined as a middle state and a leaf node is defined as an extractor node. After that, if the nodes are connected in the same way as the hierarchical structure of the data model, Security Controller can automatically construct the DFA. Therefore, the DFA can be conveniently built by investigating the link structure using the stack through a Depth-First Search, starting from the root node.

The Policy Generator uses PyangBind to construct the hierarchy of the NFI YANG data model into a Python class. This allows an XML or JSON form to be generated automatically even with updates of the NFI YANG data model. Thus, the security policy translator is able to auto-adapt to the NFI YANG data model.

6.2. Data Conversion

Security Controller requires the ability to materialize the abstract data in the high-level security policy and forward it to NSFs. Security Controller can receive endpoint information as keywords through the high-level security policy. At this time, if the endpoint information corresponding to the keyword is mapped and the query is transmitted to the NSF Database, the NSF Database can be conveniently registered with necessary information for data conversion. When a policy tries to establish a policy through the keyword, Security Controller searches for the details corresponding to the keyword registered in the NSF Database and converts the keyword into appropriate and specific data.

6.3. Policy Provisioning

This document states that a policy provisioning function is necessary to enable an I2NSF User without expert security knowledge to create policies. Policy provisioning is determined by the capability of the NSF. If the information about an NSF's capability for a policy is available to Security Controller, the probability of the selection of an appropriate NSF may increase.

Most importantly, selected NSFs may be able to perform all capabilities in the security policy. This document recommends the study of policy provisioning algorithms that are highly efficient and can satisfy all capabilities in the security policy.

7. Features of Security Policy Translator Design

First, by showing a visualized translator structure, the security manager can handle various policy changes. Translator can be shown by visualizing DFA so that the manager can easily understand the structure of Security Policy Translator.

Second, if it only keeps the hierarchy of the data model, an I2NSF User can freely create high-level policies. In the case of DFA, data extraction can be performed in the same way even if the order of input is changed. The design of the security policy translator is more flexible than the existing method that works by keeping the tag's position and order exactly.

Third, the structure of Security Policy Translator can be updated even while Security Policy Translator is operating. Because Security Policy Translator is modularized, the translator can adapt to changes in an NSF's capabilities while the I2NSF framework is running. The function of changing the translator's structure can be provided through the Registration Interface [I-D.ietf-i2nsf-registration-interface-dm].

8. Security Considerations

The data saved in the database used by the translation process MUST be kept securely. Some of the data used to convert the data from the high-level security policy to the corresponding low-level security policy may be considered private, e.g., IP addresses mapped to users. And also, alterated data in the database can cause a mistranslation and create a vulnerability in the network security that can be utilized by the attacker.

The configuration and its delivery MUST follow the security considerations discussed in [I-D.ietf-i2nsf-consumer-facing-interface-dm] and [I-D.ietf-i2nsf-nsf-facing-interface-dm].

9. IANA Considerations

This document does not require any IANA actions.

10. References

10.1. Normative References

[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>.

[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>.

[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>.

[I-D.ietf-i2nsf-consumer-facing-interface-dm] Jeong, J. P., Chung, C., Ahn, T., Kumar, R., and S. Hares, "I2NSF Consumer-Facing Interface YANG Data Model", Work in Progress, Internet-Draft, draft-ietf-i2nsf-consumer-facing-interface-dm-31, 15 May 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-consumer-facing-interface-dm-31>.

[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-registration-interface-dm]
Hyun, S., Jeong, J. P., Roh, T., Wi, S., and J. Jung-Soo, "I2NSF Registration Interface YANG Data Model for NSF Capability Registration", Work in Progress, Internet-Draft, draft-ietf-i2nsf-registration-interface-dm-26, 10 May 2023, <https://datatracker.ietf.org/doc/html/draft-ietf-i2nsf-registration-interface-dm-26>.

[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>.

10.2. Informative References

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

[Automata] Peter, L., "Formal Languages and Automata, 6th Edition", January 2016.

[Zhang-Shasha]
Zhang, K. and D. Shasha, "Simple Fast Algorithms for the Editing Distance Between Trees and Related Problems", SIAM J. Comput. https://www.researchgate.net/publication/220618233_Simple_Fast_Algorithms_for_the_Editing_Distance_Between_Trees_and_Related_Problems, 1989.

[PyangBind]
Shakir, R., "PyangBind", PyangBind https://github.com/robshakir/pyangbind, 2018.

[XML] W3C, "On Views and XML (Extensible Markup Language)", June 1999.

[XSLT] W3C, "Extensible Stylesheet Language Transformations (XSLT) Version 1.0", November 1999.

Appendix A. Mapping Information for Data Conversion

Figure 16 shows a mapping list of data fields between Consumer-Facing Interface YANG data model and NSF-Facing Interface YANG data model. Figure 16 describes the process of passing the data value to the appropriate data field of the Data Model in detail after the data conversion.

```
#policy name mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/name
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /name

#language mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/language
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /language

#resolution-strategy mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/resolution-strategy
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /resolution-strategy

#rule name mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/name
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/name

#rule priority mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/priority
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/priority

#event mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/event
/system-event
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/event/system-event

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/event
/system-alarm
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/event/system-alarm

#firewall-condition source target reference and mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/firewall/source
  -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-group
```

```
                  /("user-group" or "device-group")/name
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or "device-group")/mac-address
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ethernet
                    /source-mac-address
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or device-group")/range-ipv4-address
                  /("start" and "end")
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ipv4
                    /("source-ipv4-network" or "source-ipv4-range")
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or device-group")/range-ipv6-address
                  /("start" and "end")
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ipv6
                    /("source-ipv6-network" or "source-ipv6-range")

  #firewall-condition destination target reference and mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rule/condition
  /firewall/destination
    -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or "device-group")/name
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or "device-group")/mac-address
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ethernet
                    /destination-mac-address
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or "device-group")/range-ipv4-address
                  /("start" and "end")
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ipv4
                    /("destination-ipv4-network" or
                      "destination-ipv4-range")
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-group
                  /("user-group" or "device-group")/range-ipv6-address
                  /("start" and "end")
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/ipv6
                    /("destination-ipv6-network" or
                      "destination-ipv6-range")

  #ddos-condition threshold mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /ddos/packet-rate-threshold
    -> mapping: /ietf-i2nsf-nsf-facing-interface
```

```
                /i2nsf-security-policy/rules/condition
                /ddos/alert-packet-rate

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /ddos/packet-byte-threshold
    -> mapping: /ietf-i2nsf-nsf-facing-interface
                /i2nsf-security-policy/rules/condition
                /ddos/alert-byte-rate

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /ddos/flow-rate-threshold
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/ddos/alert-flow-rate

  #anti-virus-condition mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /anti-virus/profile
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/anti-virus/profile

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /anti-virus/exception-files
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/anti-virus/exception-files

  #payload-condition mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /payload/content
    -> reference: /ietf-i2nsf-cons-facing-interface/threat-prevention
                  /payload-content/name
      -> extract: /ietf-i2nsf-cons-facing-interface/threat-prevention
                  /payload-content/contents
                  /("content" and "depth" and "starting-point")
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition
                    /payload/content

  #voice-condition mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /voice/source-id
    -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                  /voice-group/name
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                  /voice-group/sip-id
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/voice
                    /source-voice-id
```

```
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/voice/destination-id
  -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /voice-group/name
    -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /voice-group/sip-id
      -> mapping: /ietf-i2nsf-nsf-facing-interface
                  /i2nsf-security-policy/rules/condition/voice
                  /destination-voice-id

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/voice/user-agent
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/voice/user-agent

#geographic-location mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/geographic-location/source
/("country" and "region" and "city")
  -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/("country" and "region" and "city")
    -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/range-ipv4-address
                /("start" and "end")
      -> mapping: /ietf-i2nsf-nsf-facing-interface
                  /i2nsf-security-policy/rules/condition/ipv4
                  /("source-ipv4-network" or "source-ipv4-range")
    -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/range-ipv6-address
                /("start" and "end")
      -> mapping: /ietf-i2nsf-nsf-facing-interface
                  /i2nsf-security-policy/rules/condition/ipv6
                  /("source-ipv6-network" or "source-ipv6-range")

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/geographic-location/destination
/("country" and "region" and "city")
  -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/("country" and "region" and "city")
    -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/range-ipv4-address
                /("start" and "end")
      -> mapping: /ietf-i2nsf-nsf-facing-interface
                  /i2nsf-security-policy/rules/condition/ipv4
                  /("destination-ipv4-network" or
                    "destination-ipv4-range")
    -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                /location-group/range-ipv6-address
```

```
                  /("start" and "end")
       -> mapping: /ietf-i2nsf-nsf-facing-interface
                   /i2nsf-security-policy/rules/condition/ipv4
                   /("destination-ipv6-network" or
                     "destination-ipv6-range")

  #url-condition mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /url/url-name
    -> reference: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                  /url-group/name
      -> extract: /ietf-i2nsf-cons-facing-interface/endpoint-groups
                  /url-group/url
        -> mapping: /ietf-i2nsf-nsf-facing-interface
                    /i2nsf-security-policy/rules/condition/url-category
                    /("pre-defined" or "user-defined")

  #time mapping
  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /context/time/start-date-time
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/context/time/start-date-time

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
  /context/time/end-date-time
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/context/time/end-date-time

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
  /rules/condition/context/time/period/start-time
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/context/time/period/start-time

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
  /rules/condition/context/time/period/end-time
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/context/time/period/end-time

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
  /rules/condition/context/time/period/day
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/condition/context/time/period/day

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
  /rules/condition/context/time/period/date
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules//condition/context/time/period/date
```

```
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
/rules/condition/context/time/period/month
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/time/period/month

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/
/rules/condition/context/time/frequency
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/time/frequency

#application mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/application/protocol
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/application/protocol

#device-type mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/device-type/device
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/device-type/device

#users mapping
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/users/user/id
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/users/user/id

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/users/user/name
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/users/user/name

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/users/group/id
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/users/group/id

/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/condition
/context/users/group/name
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/condition/context/users/group/name

#rule action name mapping (with Service Function Chaining)
/ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/action
/primary-action/action
  -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
              /rules/action/("packet-action" or "flow-action")
```

```
                /("ingress-action" or "egress-action")
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/action/advanced-action
                /("content-security-control" or
                  "attack-mitigation-control")

  /ietf-i2nsf-cons-facing-interface/i2nsf-cfi-policy/rules/action
  /secondary-action/log-action
    -> mapping: /ietf-i2nsf-nsf-facing-interface/i2nsf-security-policy
                /rules/action/packet-action/log-action
```

Figure 16: Mapping Information for Data Conversion

The mapping list shown in the Figure 16 shows all mapped components. This data list should be saved into the NSF Database to provide the mapping information for converting the data. It is important to produce the list automatically as the Consumer-Facing Interface and NSF-Facing Interface can be extended anytime by vendors according to the provided NSF. The Data Model Mapper in Security Policy Translator should be used to produce the mapping model information automatically.

Appendix B. 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 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)(2020-0-00395-003, Standard Development of Blockchain based Network Management Automation Technology).

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. 2022-0-01015, Development of Candidate Element Technology for Intelligent 6G Mobile Core Network).

Appendix C. Contributors

The following are coauthors of this document:

Chaehong Chung
Department of Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4106
Email: darkhong@skku.edu

Jung-Soo Park
Standards & Open Source Research Division
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu,
Daejeon
34129
Republic of Korea
Email: pjs@etri.re.kr

Younghan Kim
School of Electronic Engineering
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul
06978
Republic of Korea
Email: younghak@ssu.ac.kr

Appendix D. Changes from draft-yang-i2nsf-security-policy-translation-15

The following changes are made from draft-yang-i2nsf-security-policy-translation-15:

* This version is for only maintenance.

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Patrick Lingga
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

Jinhyuk Yang
Department of Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea
Phone: +82 10 8520 8039
Email: jin.hyuk@skku.edu