Workgroup: I2NSF Working Group
Internet-Draft:
draft-ietf-i2nsf-capability-data-model-31
Published: 14 May 2022
Intended Status: Standards Track
Expires: 15 November 2022
Authors: S. Hares, Ed. J. Jeong, Ed.
Huawei Sungkyunkwan University
J. Kim R. Moskowitz Q. Lin
Sungkyunkwan University HTT Consulting Huawei

I2NSF Capability YANG Data Model

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

Copyright Notice

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@2022-05-14.yang"

module ietf-i2nsf-capability {
  yang-version 1.1;
  namespace
    "urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability";
  prefix
    nsfcap;

  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 "2022-05-14"{
  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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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-14: 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: nsfcap
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. (., Jeong, J. (., 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-26, 19 April 2022, <https://www.ietf.org/archive/id/draft-ietf-i2nsf-nsf-facing-interface-dm-26.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.

```xml
<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.

```xml
<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-30

The following changes are made from draft-ietf-i2nsf-capability-data-model-30:

* RFC 8805 is moved to Normative Reference as RFC8805 was appropriately called out as a DOWNREF in the IETF LC with no objection from the community, and it is needed to fully explain the semantics of the YANG model.

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