I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 29, 2019

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

March 28, 2019

I2NSF Capability YANG Data Model
draft-ietf-i2nsf-capability-data-model-04

Abstract

This document defines a YANG data model for capabilities of various Network Security Functions (NSFs) in Interface to Network Security Functions (I2NSF) framework to cetrally manage capabilities of varios NSFs.

Status of This Memo

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

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

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

This Internet-Draft will expire on September 29, 2019.

Table of Contents

1. Introduction

As the industry becomes more sophisticated and network devices (e.g., Internet of Things, Self-driving vehicles, and VoIP/VoLTE smartphones), service providers have a lot of problems mentioned in [RFC8192]. To resolve these problems, [i2nsf-nsf-cap-im] specifies the information model of the capabilities of Network Security Functions (NSFs).

This document provides a data model using YANG [RFC6020][RFC7950] that defines the capabilities of NSFs to centrally manage capabilities of those security devices. The security devices can register their own capabilities into Network Operator Management (Mgmt) System (i.e., Security Controller) with this YANG data model through the registration interface [RFC8329]. With the capabilities of those security devices registered centrally, those security devices can be easily managed [RFC8329]. This YANG data model is based on the information model for I2NSF NSF capabilities [i2nsf-nsf-cap-im].

This YANG data model uses an "Event-Condition-Action" (ECA) policy model that is used as the basis for the design of I2NSF Policy described in [RFC8329] and [i2nsf-nsf-cap-im]. Rules. The "ietf-i2nsf-capability" YANG module defined in this document provides the following features:

- Definition for general capabilities of network security functions.
- Definition for event capabilities of generic network security function.
- Definition for condition capabilities of generic network security function.
- Definition for condition capabilities of advanced network security function.
- Definition for action capabilities of generic network security function.
- Definition for resolution strategy capabilities of generic network security function.
- Definition for default action capabilities of generic network security function.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119][RFC8174].

3. Terminology

This document uses the terminology described in [i2nsf-terminology][i2nsf-nsf-cap-im] [RFC8431][supa-policy-info-model]. Especially, the following terms are from [supa-policy-info-model]:

- Data Model: A data model is a representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and protocol.
- Information Model: An information model is a representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol.

3.1. Tree Diagrams

A simplified graphical representation of the data model is used in this document. The meaning of the symbols in these diagrams [RFC8340] is as follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node and "*" denotes a "list" and "leaf-list".
- Parentheses enclose choice and case nodes, and case nodes are also marked with a colon (":").
- Ellipsis ("...") stands for contents of subtrees that are not shown.

4. Overview

This section explains overview how the YANG data model can be used in I2NSF framework described in [RFC8329]. Figure 1 shows capabilities of NSFs in I2NSF Framework. As shown in this figure, Developer's Mgmt System can register NSFs with capabilities that the network security device can support. To register NSFs in this way, the Developer's Mgmt System utilizes this standardized capabilities YANG data model through registration interface. With the capabilities of those network security devices registered centrally, those security devices can be easily managed, which can resolve the a lot of problems described in [RFC8192]. The following shows use cases.

Note [i2nsf-nsf-yang] is used to configure security policy rules of generic network security functions and [i2nsf-advanced-nsf-dm] is used to configure security policy rules of advanced network security functions according to the capabilities of network security devices registed in I2NSF Framework.

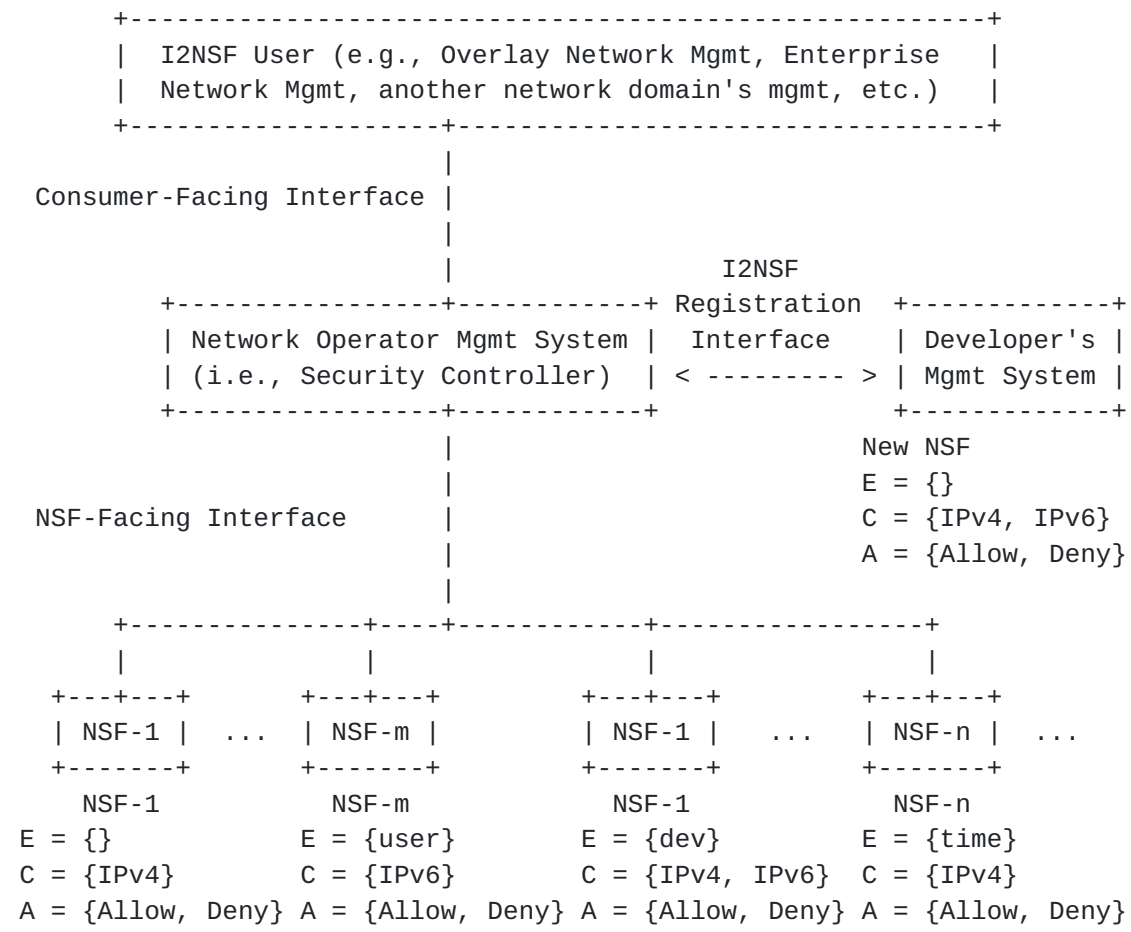

Figure 1: Capabilities of NSFs in I2NSF Framework

- If network manager wants to apply security policy rules about blocking malicious users, it is a tremendous burden to apply all of these rules to NSFs one by one. This problem can be resolved by managing the capabilities of NSFs. If network manager wants to block malicious users with IPv6, network manager sends the security policy rules about blocking the users to Network Operator Mgmt System using I2NSF user (i.e., a web browser or a software). When the Network Operator Mgmt System receives the security policy rules, it automatically sends that security policy rules to appropriate NSFs (i.e., NSF-m in Developer Mgmt System A and NSF-1 in Developer Mgmt System B) which can support the capabilities (i.e., IPv6). Therefore, I2NSF User need not consider NSFs where to apply the rules.

- If NSFs find the malicious packets, it is a tremendous burden for network manager to apply the rule about blocking the malicious packets to NSFs one by one. This problem can be resolved by

managing the capabilities of NSFs. If NSFs find the suspicious packets with IPv4, they can ask the Network Operator Mgmt System for information about the suspicious packets with IPv4. to alter specific rules and/or configurations. When the Network Operator Mgmt System receives information, it inspects the information about the suspicious packets with IPv4. If the suspicious packets are determined to be malicious packets, the Network Operator Mgmt System creates and sends the security policy rule against malicious packets to appropriate NSFs (i.e., NSF-1 in Developer Mgmt System A and NSF-1 and NSF-n in Developer Mgmt System B) which can support the capabilities (i.e., IPv4). Therefore, the new security policy rule against malicious packets can be applied to appropriate NSFs without intervention of humans.

5. YANG Tree Diagram

This section shows an YANG tree diagram of capabilities for network security functions, as defined in the [i2nsf-nsf-cap-im].

5.1. Capabilities of Network Security Function

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

```
module: ietf-i2nsf-capability
  +--rw nsf
     +--rw time-capabilities*                  enumeration
     +--rw event-capabilities
     |  +--rw system-event-capa*   identityref
     |  +--rw system-alarm-capa*   identityref
     +--rw condition-capabilities
     |  +--rw generic-nsf-capabilities
     |  |  +--rw ipv4-capa*   identityref
     |  |  +--rw ipv6-capa*   identityref
     |  |  +--rw tcp-capa*    identityref
     |  |  +--rw udp-capa*    identityref
     |  |  +--rw icmp-capa*   identityref
     |  +--rw advanced-nsf-capabilities
     |  |  +--rw antivirus-capa*    identityref
     |  |  +--rw antiddos-capa*     identityref
     |  |  +--rw ips-capa*          identityref
     |  |  +--rw url-capa*          identityref
     |  |  +--rw voip-volte-capa*   identityref
     |  +--rw context-capabilities*        identityref
     +--rw action-capabilities
     |  +--rw ingress-action-capa*   identityref
     |  +--rw egress-action-capa*    identityref
     |  +--rw log-action-capa*       identityref
     +--rw resolution-strategy-capabilities*   identityref
     +--rw default-action-capabilities*        identityref
     +--rw ipsec-method*                       identityref
```

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

This YANG tree diagram shows capabilities of network security functions.

The NSF includes NSF capabilities. The NSF capabilities include time capabilities, event capabilities, condition capabilities, action capabilities, resolution strategy capabilities, and default action capabilities.

Time capabilities are used to specify capabilities when to execute the I2NSF policy rule. The time capabilities are defined as absolute time and periodic time.

Event capabilities are used to specify capabilities how to trigger the evaluation of the condition clause of the I2NSF Policy Rule. The event capabilities are defined as system event and system alarm. The event capability can be extended according to specific vendor

condition features. The event capability is described in detail in [i2nsf-nsf-cap-im].

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 or not the set of actions in that (imperative) I2NSF policy rule can be executed or not. The condition capability is classified as condition capabilities of generic network security functions and advanced network security functions. The condition capabilities of generic network security functions are defined as IPv4 capability, IPv6 capability, tcp capability, udp capability, and icmp capability. The condition capabilities of advanced network security functions are defined as antivirus capability, antiddos capability, ips capability, http capability, and VoIP/VoLTE capability. The condition capability can be extended according to specific vendor condition features. The condition capability is described in detail in [i2nsf-nsf-cap-im].

Action capabilities is used to specify capabilities how to control and monitor 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. The action capability can be extended according to specific vendor action features. The action capability is described in detail in [i2nsf-nsf-cap-im].

Resolution strategy capabilities are used to specify capabilities how to resolve conflicts that occur between the actions of the same or different policy rules that are matched and contained in this particular NSF. The resolution strategy capabilities are defined as First Matching Rule (FMR), Last Matching Rule (LMR), Prioritized Matching Rule (PMR) with Errors (PMRE), and Prioritized Matching Rule with No Errors (PMRN). The resolution strategy capability can be extended according to specific vendor action features. The resolution strategy capability is described in detail in [i2nsf-nsf-cap-im].

Default action capabilities are used to specify capabilities how to execute I2NSF policy rule when no rule matches a packet. The default action capabilities are defined as pass, drop, reject, alert, and mirror. The default action capability can be extended according to specific vendor action features. The default action capability is described in detail in [i2nsf-nsf-cap-im].

IPsec method capabilities are used to specify capabilities how to support an Internet key exchange for the security communication. The default action capabilities are defined as ike and ikeless. The

default action capability can be extended according to specific vendor action features. The default action capability is described in detail in [draft-ietf-i2nsf-sdn-ipsec-flow-protection].

6. YANG Data Modules

6.1. I2NSF Capability YANG Data Module

This section introduces an YANG data module for capabilities of network security functions, as defined in the [i2nsf-nsf-cap-im].

<CODE BEGINS> file "ietf-i2nsf-capability@2019-03-28.yang"

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


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

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

     WG Chair: Adrian Farrel
     <mailto:Adrain@olddog.co.uk>

     WG Chair: Linda Dunbar
     <mailto:Linda.duhbar@huawei.com>

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

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

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

  description
    "This module describes a capability model
    for I2NSF devices.
```

```
  revision "2019-03-28"{
    description "Initial revision.";
    reference
      "RFC XXXX: I2NSF Capability YANG Data Model";
  }

  /*
   * Identities
   */

  identity event {
    description
      "Base identity for event of policy.";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - Event";
  }

  identity system-event-capa {
    base event;
    description
      "Identity for system event";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }

  identity system-alarm-capa {
    base event;
    description
      "Identity for system alarm";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }
```

```
  identity access-violation {
    base system-event-capa;
    description
      "Identity for access violation
      among system events";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System event";
  }

  identity configuration-change {
    base system-event-capa;
    description
      "Identity for configuration change
      among system events";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System event";
  }

  identity memory-alarm {
    base system-alarm-capa;
    description
      "Identity for memory alarm
      among system alarms";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }

  identity cpu-alarm {
    base system-alarm-capa;
    description
      "Identity for cpu alarm
      among system alarms";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }

  identity disk-alarm {
    base system-alarm-capa;
    description
      "Identity for disk alarm
      among system alarms";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
```

```
  }

  identity hardware-alarm {
    base system-alarm-capa;
    description
      "Identity for hardware alarm
      among system alarms";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }

  identity interface-alarm {
    base system-alarm-capa;
    description
      "Identity for interface alarm
      among system alarms";
    reference
      "draft-hong-i2nsf-nsf-monitoring-data-model-06
       - System alarm";
  }

  identity condition {
    description
      "Base identity for conditions of policy";
  }

  identity context-capa {
    base condition;
    description
      "Identity for capabilities of context condition";
  }

  identity acl-number {
    base context-capa;
    description
      "Identity for acl number capability
      of context condition";
  }

  identity application {
    base context-capa;
    description
      "Identity for application capability
      of context condition";
  }

  identity target {
```

```
    base context-capa;
    description
      "Identity for target capability
      of context condition";
  }

  identity user {
    base context-capa;
    description
      "Identity for user capability
      of context condition";
  }

  identity group {
    base context-capa;
    description
      "Identity for group capability
      of context condition";
  }

  identity geography {
    base context-capa;
    description
      "Identity for geography capability
      of context condition";
  }

  identity ipv4-capa {
    base condition;
    description
      "Identity for capabilities of IPv4 condition";
    reference
      "RFC 791: Internet Protocol";
  }

  identity exact-ipv4-header-length {
    base ipv4-capa;
    description
      "Identity for exact header length capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Header Length";
  }

  identity range-ipv4-header-length {
    base ipv4-capa;
    description
      "Identity for range header length capability
```

```
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Header Length";
  }

  identity ipv4-tos {
    base ipv4-capa;
    description
      "Identity for type of service capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Type of Service";
  }

  identity exact-ipv4-total-length {
    base ipv4-capa;
    description
      "Identity for exact total length capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Total Length";
  }

  identity range-ipv4-total-length {
    base ipv4-capa;
    description
      "Identity for range total length capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Total Length";
  }

  identity ipv4-id {
    base ipv4-capa;
    description
      "Identity for identification capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Identification";
  }

  identity ipv4-fragment-flags {
    base ipv4-capa;
    description
      "Identity for fragment flags capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Fragmentation  Flags";
```

```
  }

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

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

  identity exact-ipv4-ttl {
    base ipv4-capa;
    description
      "Identity for exact time to live capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Time To Live (TTL)";
  }

  identity range-ipv4-ttl {
    base ipv4-capa;
    description
      "Identity for range time to live capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Time To Live (TTL)";
  }

  identity ipv4-protocol {
    base ipv4-capa;
    description
      "Identity for protocol capability
      of IPv4 condition";
    reference
      "RFC 790: Assigned numbers - Assigned Internet
       Protocol Number
       RFC 791: Internet Protocol - Protocol";
  }
```

```
  identity exact-ipv4-address {
    base ipv4-capa;
    description
      "Identity for exact address capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Address";
  }

  identity range-ipv4-address {
    base ipv4-capa;
    description
      "Identity for range-address capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Address";
  }

  identity ipv4-ipopts {
    base ipv4-capa;
    description
      "Identity for option capability
      of IPv4 condition";
    reference
      "RFC 791: Internet Protocol - Options";
  }

  identity ipv4-sameip {
    base ipv4-capa;
    description
      "Identity for sameIP capability
      of IPv4 condition";
  }

  identity ipv4-geoip {
    base ipv4-capa;
    description
      "Identity for geography capability
      of IPv4 condition";
  }

  identity ipv6-capa {
    base condition;
    description
      "Identity for capabilities of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification";
```

```
  }

  identity ipv6-traffic-class {
    base ipv6-capa;
    description
      "Identity for traffic class capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Traffic Class";
  }

  identity exact-ipv6-flow-label {
    base ipv6-capa;
    description
      "Identity for exact flow label capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Flow Label";
  }

  identity range-ipv6-flow-label {
    base ipv6-capa;
    description
      "Identity for range flow label capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Flow Label";
  }

  identity exact-ipv6-payload-length {
    base ipv6-capa;
    description
      "Identity for exact payload length capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Payload Length";
  }

  identity range-ipv6-payload-length {
    base ipv6-capa;
    description
      "Identity for range payload length capability
      of IPv6 condition";
    reference
```

```
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Payload Length";
  }

  identity ipv6-next-header {
    base ipv6-capa;
    description
      "Identity for next header capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Next Header";
  }

  identity exact-ipv6-hop-limit {
    base ipv6-capa;
    description
      "Identity for exact hop limit capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Hop Limit";
  }

  identity range-ipv6-hop-limit {
    base ipv6-capa;
    description
      "Identity for range hop limit capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Hop Limit";
  }

  identity exact-ipv6-address {
    base ipv6-capa;
    description
      "Identity for exact address capability
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Address";
  }

  identity range-ipv6-address {
    base ipv6-capa;
    description
      "Identity for range address capability
```

```
      of IPv6 condition";
    reference
      "RFC 2460: Internet Protocol, Version 6 (IPv6)
      Specification - Address";
  }

  identity tcp-capa {
    base condition;
    description
      "Identity for capabilities of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol";
  }

  identity exact-tcp-port-num {
    base tcp-capa;
    description
      "Identity for exact port number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Port Number";
  }

  identity range-tcp-port-num {
    base tcp-capa;
    description
      "Identity for range port number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Port Number";
  }

  identity exact-tcp-seq-num {
    base tcp-capa;
    description
      "Identity for exact sequence number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Sequence Number";
  }

  identity range-tcp-seq-num {
    base tcp-capa;
    description
      "Identity for range sequence number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Sequence Number";
```

```
  }

  identity exact-tcp-ack-num {
    base tcp-capa;
    description
      "Identity for exact acknowledgement number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Acknowledgement Number";
  }

  identity range-tcp-ack-num {
    base tcp-capa;
    description
      "Identity for range acknowledgement number capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Acknowledgement Number";
  }

  identity exact-tcp-window-size {
    base tcp-capa;
    description
      "Identity for exact window size capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Window Size";
  }

  identity range-tcp-window-size {
    base tcp-capa;
    description
      "Identity for range window size capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Window Size";
  }

  identity tcp-flags {
    base tcp-capa;
    description
      "Identity for flags capability
      of tcp condition";
    reference
      "RFC 793: Transmission Control Protocol - Flags";
  }

  identity udp-capa {
```

```
    base condition;
    description
      "Identity for capabilities of udp condition";
    reference
      "RFC 768: User Datagram Protocol";
  }

  identity exact-udp-port-num {
    base udp-capa;
    description
      "Identity for exact port number capability
      of udp condition";
    reference
      "RFC 768: User Datagram Protocol - Port Number";
  }

  identity range-udp-port-num {
    base udp-capa;
    description
      "Identity for range port number capability
      of udp condition";
    reference
      "RFC 768: User Datagram Protocol - Port Number";
  }

  identity exact-udp-total-length {
    base udp-capa;
    description
      "Identity for exact total-length capability
      of udp condition";
    reference
      "RFC 768: User Datagram Protocol - Total Length";
  }

  identity range-udp-total-length {
    base udp-capa;
    description
      "Identity for range total-length capability
      of udp condition";
    reference
      "RFC 768: User Datagram Protocol - Total Length";
  }

  identity icmp-capa {
    base condition;
    description
      "Identity for capabilities of icmp condition";
    reference
```

```
      "RFC 792: Internet Control Message Protocol";
  }

  identity icmp-type {
    base icmp-capa;
    description
      "Identity for icmp type capability
      of icmp condition";
    reference
      "RFC 792: Internet Control Message Protocol";
  }

  identity url-capa {
    base condition;
    description
      "Identity for capabilities of url condition";
  }

  identity pre-defined {
    base url-capa;
    description
      "Identity for pre-defined capabilities of
       url condition";
  }

  identity user-defined {
    base url-capa;
    description
      "Identity for user-defined capabilities of
       url condition";
  }

  identity log-action-capa {
    description
      "Identity for capabilities of log action";
  }

  identity rule-log {
    base log-action-capa;
    description
      "Identity for rule log capability
      of log action";
  }

  identity session-log {
    base log-action-capa;
    description
      "Identity for session log capability
```

```
      of log action";
  }

  identity ingress-action-capa {
    description
      "Identity for capabilities of ingress action";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Action";
  }

  identity egress-action-capa {
    description
      "Base identity for egress action";
  }

  identity default-action-capa {
    description
      "Identity for capabilities of default action";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Default action";
  }

  identity pass {
    base ingress-action-capa;
    base egress-action-capa;
    base default-action-capa;
    description
      "Identity for pass";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Actions and
       default action";
  }

  identity drop {
    base ingress-action-capa;
    base egress-action-capa;
    base default-action-capa;
    description
      "Identity for drop";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Actions and
       default action";
  }
```

```
  identity reject {
    base ingress-action-capa;
    base egress-action-capa;
    base default-action-capa;
    description
      "Identity for reject";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Actions and
       default action";
  }

  identity alert {
    base ingress-action-capa;
    base egress-action-capa;
    base default-action-capa;
    description
      "Identity for alert";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Actions and
       default action";
  }

  identity mirror {
    base ingress-action-capa;
    base egress-action-capa;
    base default-action-capa;
    description
      "Identity for mirror";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Actions and
       default action";
  }

  identity invoke-signaling {
    base egress-action-capa;
    description
      "Identity for invoke signaling";
  }

  identity tunnel-encapsulation {
    base egress-action-capa;
    description
      "Identity for tunnel encapsulation";
  }
```

```
  identity forwarding {
    base egress-action-capa;
    description
      "Identity for forwarding";
  }

  identity redirection {
    base egress-action-capa;
    description
      "Identity for redirection";
  }

  identity resolution-strategy-capa {
    description
      "Base identity for resolution strategy";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity fmr {
    base resolution-strategy-capa;
    description
      "Identity for First Matching Rule (FMR)";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity lmr {
    base resolution-strategy-capa;
    description
      "Identity for Last Matching Rule (LMR)";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity pmr {
    base resolution-strategy-capa;
    description
      "Identity for Prioritized Matching Rule (PMR)";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity pmre {
```

```
    base resolution-strategy-capa;
    description
      "Identity for Prioritized Matching Rule
      with Errors (PMRE)";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity pmrn {
    base resolution-strategy-capa;
    description
      "Identity for Prioritized Matching Rule
      with No Errors (PMRN)";
    reference
      "draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Resolution Strategy";
  }

  identity advanced-nsf-capa {
    description
      "Base identity for advanced
      network security function capabilities";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Differences from ACL Data Models
       draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller";
  }

  identity antivirus-capa {
    base advanced-nsf-capa;
    description
      "Identity for antivirus capabilities";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Differences from ACL Data Models
       draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antivirus";
  }

  identity antiddos-capa {
    base advanced-nsf-capa;
    description
      "Identity for antiddos capabilities";
    reference
```

```
      "RFC 8329: Framework for Interface to Network Security
       Functions - Differences from ACL Data Models
       draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity ips-capa {
    base advanced-nsf-capa;
    description
      "Identity for IPS capabilities";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Differences from ACL Data Models
       draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Intrusion Prevention System";
  }

  identity voip-volte-capa {
    base advanced-nsf-capa;
    description
      "Identity for VoIP/VoLTE capabilities";
    reference
      "RFC 3261: SIP: Session Initiation Protocol
       RFC 8329: Framework for Interface to Network Security
       Functions - Differences from ACL Data Models
       draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller";
  }

  identity detect {
    base antivirus-capa;
    description
      "Identity for detect capabilities
      of antivirus";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antivirus";
  }

  identity exception-application {
    base antivirus-capa;
    description
      "Identity for exception application capabilities
      of antivirus";
```

```
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antivirus";
  }

  identity exception-signature {
    base antivirus-capa;
    description
      "Identity for exception signature capabilities
      of antivirus";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antivirus";
  }

  identity whitelists {
    base antivirus-capa;
    description
      "Identity for whitelists capabilities
      of antivirus";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antivirus";
  }

  identity syn-flood-action {
    base antiddos-capa;
    description
      "Identity for syn flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity udp-flood-action {
    base antiddos-capa;
    description
      "Identity for udp flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
```

```
  }

  identity http-flood-action {
    base antiddos-capa;
    description
      "Identity for http flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity https-flood-action {
    base antiddos-capa;
    description
      "Identity for https flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity dns-request-flood-action {
    base antiddos-capa;
    description
      "Identity for dns request flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity dns-reply-flood-action {
    base antiddos-capa;
    description
      "Identity for dns reply flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity icmp-flood-action {
    base antiddos-capa;
```

```
    description
      "Identity for icmp flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity sip-flood-action {
    base antiddos-capa;
    description
      "Identity for sip flood action capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity detect-mode {
    base antiddos-capa;
    description
      "Identity for detect mode capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity baseline-learn {
    base antiddos-capa;
    description
      "Identity for baseline learn capabilities
      of antiddos";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Antiddos";
  }

  identity signature-set {
    base ips-capa;
    description
      "Identity for signature set capabilities
      of IPS";
    reference
```

```
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Intrusion Prevention System";
  }

  identity ips-exception-signature {
    base ips-capa;
    description
      "Identity for ips exception signature capabilities
      of IPS";
    reference
      "draft-dong-i2nsf-asf-config-01: Configuration of
       Advanced Security Functions with I2NSF Security
       Controller - Intrusion Prevention System";
  }

  identity voice-id {
    base voip-volte-capa;
    description
      "Identity for voice-id capabilities
      of VoIP/VoLTE";
    reference
      "RFC 3261: SIP: Session Initiation Protocol";
  }

  identity user-agent {
    base voip-volte-capa;
    description
      "Identity for user agent capabilities
      of VoIP/VoLTE";
    reference
      "RFC 3261: SIP: Session Initiation Protocol";
  }

  identity ipsec-capa {
    description
      "Base identity for an IPsec";
  }

  identity ike {
    base ipsec-capa;
    description
      "Identity for an IKE";
  }

  identity ikeless {
    base ipsec-capa;
    description
```

```
      "Identity for an IKEless";
  }



  /*
   *  Grouping
   */

  grouping nsf-capabilities {
    description
      "Capabilities of network security funtion";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - I2NSF Flow Security Policy Structure
       draft-ietf-i2nsf-capability-04: Information Model
       of NSFs Capabilities - Capability Information Model Design";

    leaf-list time-capabilities {
      type enumeration {
        enum absolute-time {
          description
            "Capabilities of absolute time.
             If network security function has the absolute time
             capability, the network security function
             supports rule execution according to absolute time.";
        }
        enum periodic-time {
          description
            "Capabilities of periodic time.
             If network security function has the periodic time
             capability, the network security function
             supports rule execution according to periodic time.";
        }
      }
      description
        "This is capabilities for time";
    }

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

      reference
```

```
        "RFC 8329: Framework for Interface to Network Security
         Functions - I2NSF Flow Security Policy Structure
         draft-ietf-i2nsf-capability-04: Information Model
         of NSFs Capabilities - Design Principles and ECA
         Policy Model Overview
         draft-hong-i2nsf-nsf-monitoring-data-model-06: A YANG
         Data Model for Monitoring I2NSF Network Security
         Functions - System Alarm and System Events";

      leaf-list system-event-capa {
        type identityref {
          base system-event-capa;
        }
        description
          "Capabilities for a system event";
      }

      leaf-list system-alarm-capa {
        type identityref {
          base system-alarm-capa;
        }
        description
          "Capabilities for a system alarm";
      }
    }

    container condition-capabilities {
      description
        "Capabilities of conditions.";

      container generic-nsf-capabilities {
        description
          "Capabilities of conditions.
           If a network security function has
           the condition capabilities, the network security function
           supports rule execution according to conditions of IPv4,
           IPv6, foruth layer, ICMP, and payload.";
        reference
          "RFC  791: Internet Protocol
           RFC  792: Internet Control Message Protocol
           RFC  793: Transmission Control Protocol
           RFC 2460: Internet Protocol, Version 6 (IPv6)
           Specification - Next Header
           RFC 8329: Framework for Interface to Network Security
           Functions - I2NSF Flow Security Policy Structure
           draft-ietf-i2nsf-capability-04: Information Model
           of NSFs Capabilities - Design Principles and ECA Policy
           Model Overview";
```

```
        leaf-list ipv4-capa {
          type identityref {
            base ipv4-capa;
          }
          description
            "Capabilities for an IPv4 packet";
          reference
            "RFC 791: Internet Protocol";
        }

        leaf-list ipv6-capa {
          type identityref {
            base ipv6-capa;
          }
          description
            "Capabilities for an IPv6 packet";
          reference
            "RFC 2460: Internet Protocol, Version 6 (IPv6)
             Specification - Next Header";
        }

        leaf-list tcp-capa {
          type identityref {
            base tcp-capa;
          }
          description
            "Capabilities for a tcp packet";
          reference
            "RFC 793: Transmission Control Protocol";
        }

        leaf-list udp-capa {
          type identityref {
            base udp-capa;
          }
          description
            "Capabilities for an udp packet";
          reference
            "RFC 768: User Datagram Protocol";
        }

        leaf-list icmp-capa {
          type identityref {
            base icmp-capa;
          }
          description
            "Capabilities for an ICMP packet";
          reference
```

```
            "RFC 2460: Internet Protocol, Version 6 (IPv6) ";
        }
      }

      container advanced-nsf-capabilities {
        description
          "Capabilities of advanced network security functions,
           such as anti virus, anti DDoS, IPS, and VoIP/VoLTE.";
        reference
          "RFC 8329: Framework for Interface to Network Security
           Functions - Differences from ACL Data Models
           draft-dong-i2nsf-asf-config-01: Configuration of
           Advanced Security Functions with I2NSF Security
           Controller";

        leaf-list antivirus-capa {
          type identityref {
            base antivirus-capa;
          }
          description
            "Capabilities for an antivirus";
          reference
            "draft-dong-i2nsf-asf-config-01: Configuration of
             Advanced Security Functions with I2NSF Security
             Controller";
        }

        leaf-list antiddos-capa {
          type identityref {
            base antiddos-capa;
          }
          description
            "Capabilities for an antiddos";
          reference
            "draft-dong-i2nsf-asf-config-01: Configuration of
             Advanced Security Functions with I2NSF Security
             Controller";
        }

        leaf-list ips-capa {
          type identityref {
            base ips-capa;
          }
          description
            "Capabilities for an ips";
          reference
            "draft-dong-i2nsf-asf-config-01: Configuration of
             Advanced Security Functions with I2NSF Security
```

```
            Controller";
        }

        leaf-list url-capa {
          type identityref {
            base url-capa;
          }
          description
            "Capabilities for a url category";
          reference
            "draft-dong-i2nsf-asf-config-01: Configuration of
             Advanced Security Functions with I2NSF Security
             Controller";
        }

        leaf-list voip-volte-capa {
          type identityref {
            base voip-volte-capa;
         }
          description
            "Capabilities for a voip and volte";
          reference
            "draft-dong-i2nsf-asf-config-01: Configuration of
             Advanced Security Functions with I2NSF Security
             Controller";
        }
      }

      leaf-list context-capabilities {
        type identityref {
          base context-capa;
        }
        description
          "Capabilities for a context security";
      }

    }
    container action-capabilities {
      description
        "Capabilities of actions.
         If network security function has
         the action capabilities, the network security function
         supports rule execution according to actions.";

      leaf-list ingress-action-capa {
        type identityref {
          base ingress-action-capa;
        }
```

```
        description
          "Capabilities for an action";
      }

      leaf-list egress-action-capa {
        type identityref {
          base egress-action-capa;
        }
        description
          "Capabilities for an egress action";
      }

      leaf-list log-action-capa {
        type identityref {
          base log-action-capa;
        }
        description
          "Capabilities for a log action";
      }
    }

    leaf-list resolution-strategy-capabilities {
      type identityref {
        base resolution-strategy-capa;
      }
      description
        "Capabilities for a resolution strategy.
        The resolution strategies can be used to
        specify how to resolve conflicts that occur between
        the actions of the same or different policy rules that
        are matched and contained in this particular NSF";
      reference
        "draft-ietf-i2nsf-capability-04: Information Model
         of NSFs Capabilities - Resolution strategy";
    }

    leaf-list default-action-capabilities {
      type identityref {
        base default-action-capa;
      }
      description
        "Capabilities for a default action.
         A default action is used to execute I2NSF policy rule
         when no rule matches a packet. The default action is
         defined as pass, drop, reject, alert, and mirror.";
      reference
        "draft-ietf-i2nsf-capability-04: Information Model
         of NSFs Capabilities - Default action";
```

```
    }

    leaf-list ipsec-method {
      type identityref {
        base ipsec-capa;
      }
      description
        "Capabilities for an IPsec method";
      reference
        " draft-ietf-i2nsf-sdn-ipsec-flow-protection-04";
    }
  }


  /*
   * Data nodes
   */

  container nsf {
    description
      "The list of capabilities of
       network security function";
    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]:

URI: urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability

Registrant Contact: The IESG.

XML: N/A; the requested URI is an XML namespace.

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

name: ietf-i2nsf-capability

namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability

prefix: iicapa

reference: RFC XXXX

8. Security Considerations

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

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

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC6020] Bjorklund, M., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, October 2010.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <https://www.rfc-editor.org/info/rfc6087>.

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

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

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

[RFC7950] Bjorklund, M., "The YANG 1.1 Data Modeling Language", RFC 7950, August 2016.

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

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

[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, July 2017.

[RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, February 2018.

[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>.

[RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", RFC RFC8431, September 2018.

[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>.

9.2. Informative References

[draft-ietf-i2nsf-sdn-ipsec-flow-protection] Marin-Lopez, R., Lopez-Millan, G., and F. Pereniguez-Garcia, "Software-Defined Networking (SDN)-based IPsec Flow Protection", draft-ietf-i2nsf-sdn-ipsec-flow-protection-04 (work in progress), March 2019.

[i2nsf-advanced-nsf-dm]
Pan, W. and L. Xia, "Configuration of Advanced Security Functions with I2NSF Security Controller", draft-dong-i2nsf-asf-config-01 (work in progress), October 2018.

[i2nsf-nsf-cap-im]
Xia, L., Strassner, J., Basile, C., and D. Lopez, "Information Model of NSFs Capabilities", draft-ietf-i2nsf-capability-04 (work in progress), October 2018.

[i2nsf-nsf-yang]
Kim, J., Jeong, J., Park, J., Hares, S., and Q. Lin, "I2NSF Network Security Function-Facing Interface YANG Data Model", draft-ietf-i2nsf-nsf-facing-interface-dm-04 (work in progress), March 2019.

[i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-07 (work in progress), January 2019.

[supa-policy-info-model]
Strassner, J., Halpern, J., and S. Meer, "Generic Policy Information Model for Simplified Use of Policy Abstractions (SUPA)", draft-ietf-supa-generic-policy-info-model-03 (work in progress), May 2017.

Appendix A. Changes from draft-ietf-i2nsf-capability-data-model-03

The following changes are made from draft-ietf-i2nsf-capability-data-model-03:

- We added a leaf-list for IPsec method capabilities (e.g., ike and ikeless).
- We changed http capa fields to url category capa fields.
- We added context capa fields (e.g., acl number, application, target, users, group, and geography).

Appendix B. Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning).

Appendix C. Contributors

This document is made by the group effort of I2NSF working group. Many people actively contributed to this document. The following are considered co-authors:

- Hyoungshick Kim (Sungkyunkwan University)
- Daeyoung Hyun (Sungkyunkwan University)
- Dongjin Hong (Sungkyunkwan University)
- Liang Xia (Huawei)
- Jung-Soo Park (ETRI)
- Tae-Jin Ahn (Korea Telecom)
- Se-Hui Lee (Korea Telecom)

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

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

Jaehoon Paul Jeong
Department of Software
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 299 4957
Fax: +82 31 290 7996
EMail: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Jinyong Tim Kim
Department of 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
USA

Phone: +1-248-968-9809
EMail: rgm@htt-consult.com

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen, Guangdong 518129
China

EMail: linqiushi@huawei.com