I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 26, 2021

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

August 25, 2020

I2NSF Capability YANG Data Model
draft-ietf-i2nsf-capability-data-model-07

Abstract

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

Status of This Memo

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

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

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

This Internet-Draft will expire on February 26, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (https://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 2
2. Requirements Language 3
3. Terminology 3
3.1. Tree Diagrams 4
4. Overview 4
5. YANG Tree Diagram 6
5.1. Network Security Function (NSF) Capabilities 6
6. YANG Data Modules 9
6.1. I2NSF Capability YANG Data Module 9
7. IANA Considerations 38
8. Security Considerations 39
9. References 40
9.1. Normative References 40
9.2. Informative References 42
Appendix A. Configuration Examples 43
A.1. Example 1: Registration for Capabilities of General Firewall 43
A.2. Example 2: Registration for Capabilities of Time based Firewall 44
A.3. Example 3: Registration for Capabilities of Web Filter . 45
A.4. Example 4: Registration for Capabilities of VoIP/VoLTE Filter 45
A.5. Example 5: Registration for Capabilities of HTTP and HTTPS Flood Mitigation 46
Appendix B. Acknowledgments 47
Appendix C. Contributors 47
Authors' Addresses 48

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 described in [RFC8192]. To resolve these problems, [draft-ietf-i2nsf-capability] specifies the information model of the capabilities of Network Security Functions (NSFs).

This document provides a YANG data model [RFC6020][RFC7950] that defines the capabilities of NSFs to centrally manage the capabilities of those security devices. The security devices 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 capabilities of those security devices maintained centrally, those security devices can be more easily managed [RFC8329]. This YANG data model is based on the information model for I2NSF NSF capabilities [draft-ietf-i2nsf-capability].

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 [draft-ietf-i2nsf-capability]. 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 functions.
- Definition for condition capabilities of generic network security functions.
- Definition for condition capabilities of advanced network security functions.
- Definition for action capabilities of generic network security functions.
- Definition for resolution strategy capabilities of generic network security functions.
- Definition for default action capabilities of generic network security functions.

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 [draft-ietf-i2nsf-capability][RFC8431]. Especially, the following terms are from [RFC3444]:

- 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.

- o 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 is referred from [RFC8340].

4. Overview

This section provides as 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, an NSF Developer's Management System can register NSFs and the capabilities that the network security device can support. To register NSFs in this way, the Developer's Management System utilizes this standardized capability YANG data model through the I2NSF Registration Interface [draft-ietf-i2nsf-registration-interface-dm]. That is, this Registration Interface uses the YANG module described in this document to describe the capability of a network security function that is registered with the Security Controller. With the capabilities of those network security devices maintained centrally, those security devices can be more easily managed, which can resolve many of the problems described in [RFC8192].

In Figure 1, a new NSF at a Developer's Management Systems 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 the NSF-Facing Interface is used to configure the security policy rules of the generic network security functions [draft-ietf-i2nsf-nsf-facing-interface-dm], and The configuration of advanced security functions over the NSF-Facing Interface is used to configure the security policy rules of advanced network security functions (e.g., anti-virus and anti-DDoS attack), respectively, according to the capabilities of NSFs registered with the I2NSF Framework.

```
       +-----------------------------------------------------+
       |  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 = {time}
 C = {IPv4}        C = {IPv6}        C = {IPv4, IPv6}  C = {IPv4}
 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.

- o If a network manager 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 in this document.

- o If a network administrator wants to block malicious users for IPv6 traffic, he sends a security policy rule to block the users to the Network Operator Management System using the I2NSF User (i.e., web application).

- o When the Network Operator Management System receives the security policy rule, it automatically sends that security policy rules 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 NSFs where the rule is applied.

- o If NSFs encounter the suspicious 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 [draft-ietf-i2nsf-capability].

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 time 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 [draft-ietf-i2nsf-capability].

```
module: ietf-i2nsf-capability
  +--rw nsf* [nsf-name]
     +--rw nsf-name            string
     +--rw time-capabilities*                  enumeration
     +--rw event-capabilities
     |  +--rw system-event-capability*   identityref
     |  +--rw system-alarm-capability*   identityref
     +--rw condition-capabilities
     |  +--rw generic-nsf-capabilities
     |  |  +--rw ipv4-capability*   identityref
     |  |  +--rw icmp-capability*   identityref
     |  |  +--rw ipv6-capability*   identityref
     |  |  +--rw icmpv6-capability*   identityref
     |  |  +--rw tcp-capability*    identityref
     |  |  +--rw udp-capability*    identityref
     |  +--rw advanced-nsf-capabilities
     |  |  +--rw anti-virus-capability*    identityref
     |  |  +--rw anti-ddos-capability*     identityref
     |  |  +--rw ips-capability*          identityref
     |  |  +--rw url-capability*          identityref
     |  |  +--rw voip-volte-capability*   identityref
     |  +--rw context-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
     +--rw ipsec-method*                       identityref
```

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

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. The absolute time means the exact time to start or end. The periodic time means repeated time like day, week, or month. See Section 3.4.6 (Capability Algebra) in [draft-ietf-i2nsf-capability] for more information about the time-based condition (e.g., time period) in the capability algebra.

Event capabilities are used to specify the capabilities that describe the 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. See Section 3.1 (Design Principles and ECA

Policy Model Overview) in [draft-ietf-i2nsf-capability] for more information about the event in the ECA policy model.

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. The condition capabilities are classified in terms 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 anti-virus capability, anti-DDoS capability, IPS capability, HTTP capability, and VoIP/VoLTE capability. See Section 3.1 (Design Principles and ECA Policy Model Overview) in [draft-ietf-i2nsf-capability] for more information about the condition in the ECA policy model. Also, see Section 3.4.3 (I2NSF Condition Clause Operator Types) in [draft-ietf-i2nsf-capability] for more information about the operator types in an I2NSF condition clause.

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 (Design Principles and ECA Policy Model Overview) in [draft-ietf-i2nsf-capability] 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) 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 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), Prioritized Matching Rule with Errors (PMRE), and Prioritized Matching Rule with No Errors (PMRN). See Section 3.4.2 (Conflict, Resolution Strategy and Default Action) in [draft-ietf-i2nsf-capability] 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, alert, and mirror. See Section 3.4.2 (Conflict, Resolution Strategy and Default Action) in [draft-ietf-i2nsf-capability] for more information about the default action.

IPsec method capabilities are used to specify capabilities of how to support an Internet Key Exchange (IKE) for the security communication. The default action capabilities are defined as IKE or IKE-less. See [draft-ietf-i2nsf-sdn-ipsec-flow-protection] for more information about the SDN-based IPsec flow protection in I2NSF.

6. YANG Data Modules

6.1. I2NSF Capability YANG Data Module

This section introduces a YANG data module for network security functions capabilities, as defined in the [draft-ietf-i2nsf-capability].

```
<CODE BEGINS> file "ietf-i2nsf-capability@2020-08-25.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: <http://tools.ietf.org/wg/i2nsf>
     WG List: <mailto:i2nsf@ietf.org>

     WG Chair: Linda Dunbar
     <mailto:ldunbar@futurewei.com>

     WG Chair: Yoav Nir
     <mailto:ynir.ietf@gmail.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.

    Copyright (c) 2020 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 Simplified BSD License
    set forth in Section 4.c of the IETF Trust's Legal Provisions
    Relating to IETF Documents
    (http://trustee.ietf.org/license-info).

    This version of this YANG module is part of RFC 8341; see
    the RFC itself for full legal notices.";

  revision "2020-08-25"{
    description "Initial revision.";
    reference
      "RFC XXXX: I2NSF Capability YANG Data Model";
  }

  /*
   * Identities
   */

  identity event {
    description
      "Base identity for I2NSF policy events.";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - Event";
  }

  identity system-event-capability {
    base event;
    description
      "Identity for system events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity system-alarm-capability {
    base event;
```

```
    description
      "Identity for system alarms";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity access-violation {
    base system-event-capability;
    description
      "Identity for access violation events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System event";
  }

  identity configuration-change {
    base system-event-capability;
    description
      "Identity for configuration change events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System event";
  }

  identity memory-alarm {
    base system-alarm-capability;
    description
      "Identity for memory alarm events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity cpu-alarm {
    base system-alarm-capability;
    description
      "Identity for CPU alarm events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity disk-alarm {
    base system-alarm-capability;
    description
      "Identity for disk alarm events";
    reference
```

```
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity hardware-alarm {
    base system-alarm-capability;
    description
      "Identity for hardware alarm events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

  identity interface-alarm {
    base system-alarm-capability;
    description
      "Identity for interface alarm events";
    reference
      "draft-ietf-i2nsf-nsf-monitoring-data-model-03
       - System alarm";
  }

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

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

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

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

  identity target {
    base context-capability;
    description
```

```
      "Identity for target condition capability";
  }

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

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

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

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

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

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

  identity ipv4-tos {
```

```
    base ipv4-capability;
    description
      "Identity for IPv4 Type-Of-Service (TOS)
      condition capability";
    reference
      "RFC 791: Internet Protocol - Type of Service";
  }

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

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

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

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

  identity exact-ipv4-fragment-offset {
    base ipv4-capability;
    description
      "Identity for exact-match IPv4 fragment offset
      condition capability";
    reference
```

```
      "RFC 791: Internet Protocol - Fragmentation Offset";
  }

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

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

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

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

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

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

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

  identity ipv4-geo-ip {
    base ipv4-capability;
    description
      "Identity for geography condition capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model
       of NSFs Capabilities - Geo-IP";
  }

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

  identity ipv6-traffic-class {
    base ipv6-capability;
    description
      "Identity for IPv6 traffic class
      condition capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Traffic Class";
  }

  identity exact-ipv6-flow-label {
    base ipv6-capability;
    description
```

```
      "Identity for exact-match IPv6 flow label
      condition capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Flow Label";
  }

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

  identity exact-ipv6-payload-length {
    base ipv6-capability;
    description
      "Identity for exact-match IPv6 payload length
      condition capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Payload Length";
  }

  identity range-ipv6-payload-length {
    base ipv6-capability;
    description
      "Identity for range-match IPv6 payload length
      condition capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Payload Length";
  }

  identity ipv6-next-header {
    base ipv6-capability;
    description
      "Identity for IPv6 next header condition capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Next Header";
  }

  identity exact-ipv6-hop-limit {
    base ipv6-capability;
```

```
    description
      "Identity for exact-match IPv6 hop limit condition
      capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Hop Limit";
  }

  identity range-ipv6-hop-limit {
    base ipv6-capability;
    description
      "Identity for range-match IPv6 hop limit condition
      capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Hop Limit";
  }

  identity exact-ipv6-address {
    base ipv6-capability;
    description
      "Identity for exact-match IPv6 address condition
       capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Address";
  }

  identity range-ipv6-address {
    base ipv6-capability;
    description
      "Identity for range-match IPv6 address condition
      capability";
    reference
      "RFC 8200: Internet Protocol, Version 6 (IPv6)
      Specification - Address";
  }

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

  identity exact-tcp-port-num {
    base tcp-capability;
```

```
    description
      "Identity for exact-match TCP port number condition
       capability";
    reference
      "RFC 793: Transmission Control Protocol - Port Number";
  }

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

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

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

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

  identity range-tcp-ack-num {
    base tcp-capability;
    description
      "Identity for range-match TCP acknowledgement number condition
       capability";
```

```
    reference
      "RFC 793: Transmission Control Protocol - Acknowledgement Number";
  }

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

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

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

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

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

  identity range-udp-port-num {
    base udp-capability;
    description
      "Identity for range-match UDP port number condition capability";
```

```
    reference
      "RFC 768: User Datagram Protocol - Port Number";
  }

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

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

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

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

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

  identity icmpv6-type {
    base icmpv6-capability;
```

```
    description
      "Identity for ICMPv6 type condition capability";
    reference
      "RFC 4443: Internet Control Message Protocol (ICMPv6)
       for the Internet Protocol Version 6 (IPv6) Specification
       - ICMPv6";
  }

  identity url-capability {
    base condition;
    description
      "Identity for URL condition capability";
  }

  identity pre-defined {
    base url-capability;
    description
      "Identity for URL pre-defined condition capability";
  }

  identity user-defined {
    base url-capability;
    description
      "Identity for URL user-defined condition capability";
  }

  identity log-action-capability {
    description
      "Identity for log-action capability";
  }

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

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

  identity ingress-action-capability {
    description
      "Identity for ingress-action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
```

```
       Functions - Ingress action";
  }

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

  identity default-action-capability {
    description
      "Identity for default-action capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Default action";
  }

  identity pass {
    base ingress-action-capability;
    base egress-action-capability;
    base default-action-capability;
    description
      "Identity for pass action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and pass actions
           draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Actions and default action";
  }

  identity drop {
    base ingress-action-capability;
    base egress-action-capability;
    base default-action-capability;
    description
      "Identity for drop action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and drop actions
       draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Actions and default action";
  }

  identity alert {
    base ingress-action-capability;
    base egress-action-capability;
```

```
    base default-action-capability;
    description
      "Identity for alert action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and alert actions
       draft-ietf-i2nsf-nsf-monitoring-data-model-03: I2NSF
       NSF Monitoring YANG Data Model - Alarm (i.e., alert)
       draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Actions and default action";
  }

  identity mirror {
    base ingress-action-capability;
    base egress-action-capability;
    base default-action-capability;
    description
      "Identity for mirror action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Ingress, egress, and mirror actions
       draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Actions and default action";
  }

  identity invoke-signaling {
    base egress-action-capability;
    description
      "Identity for invoke signaling action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Invoke-signaling action";
  }

  identity tunnel-encapsulation {
    base egress-action-capability;
    description
      "Identity for tunnel encapsulation action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Tunnel-encapsulation action";
  }

  identity forwarding {
    base egress-action-capability;
    description
      "Identity for forwarding action capability";
    reference
```

```
      "RFC 8329: Framework for Interface to Network Security
       Functions - Forwarding action";
  }

  identity redirection {
    base egress-action-capability;
    description
      "Identity for redirection action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Redirection action";
  }

  identity resolution-strategy-capability {
    description
      "Base identity for resolution strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Resolution Strategy";
  }

  identity fmr {
    base resolution-strategy-capability;
    description
      "Identity for First Matching Rule (FMR) resolution
       strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Resolution Strategy";
  }

  identity lmr {
    base resolution-strategy-capability;
    description
      "Identity for Last Matching Rule (LMR) resolution
       strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Resolution Strategy";
  }

  identity pmr {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule (PMR) resolution
       strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of
```

```
       NSFs Capabilities - Resolution Strategy";
  }

  identity pmre {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule with Errors (PMRE)
       resolution strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of NSFs
      Capabilities - Resolution Strategy";
  }

  identity pmrn {
    base resolution-strategy-capability;
    description
      "Identity for Prioritized Matching Rule with No Errors (PMRN)
       resolution strategy capability";
    reference
      "draft-ietf-i2nsf-capability-05: Information Model of NSFs
       Capabilities - Resolution Strategy";
  }

  identity advanced-nsf-capability {
    description
      "Base identity for advanced Network Security Function (NSF)
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF capability";
  }

  identity anti-virus-capability {
    base advanced-nsf-capability;
    description
      "Identity for advanced NSF Anti-Virus capability";
       reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-Virus capability";
  }

  identity anti-ddos-capability {
    base advanced-nsf-capability;
    description
      "Identity for advanced NSF Anti-DDoS attack capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Attack capability";
```

```
  }

  identity ips-capability {
    base advanced-nsf-capability;
    description
      "Identity for advanced NSF Intrusion Prevention System
      (IPS) capabilities";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF IPS capability";
  }

  identity voip-volte-capability {
    base advanced-nsf-capability;
    description
      "Identity for advanced NSF VoIP/VoLTE capability";
    reference
      "RFC 3261: SIP: Session Initiation Protocol
       RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF VoIP/VoLTE capability";
  }

  identity detect {
    base anti-virus-capability;
    description
      "Identity for advanced NSF Anti-Virus Detection capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-Virus Detection capability";
  }

  identity exception-application {
    base anti-virus-capability;
    description
      "Identity for advanced NSF Anti-Virus Exception Application
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-Virus Exception Application
       capability";
  }

  identity exception-signature {
    base anti-virus-capability;
    description
      "Identity for advanced NSF Anti-Virus Exception Signature
       capability";
    reference
```

```
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-Virus Exception Signature
       capability";
  }

  identity whitelists {
    base anti-virus-capability;
    description
      "Identity for advanced NSF Anti-Virus Whitelists capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-Virus Whitelists capability";
  }

  identity syn-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS SYN Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS SYN Flood Action
       capability";
  }

  identity udp-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS UDP Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS UDP Flood Action
       capability";
  }

  identity http-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF anti-DDoS HTTP Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS HTTP Flood Action
       capability";
  }

  identity https-flood-action {
```

```
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS HTTPS Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS HTTPS Flood Action
       capability";
  }

  identity dns-request-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS DNS Request Flood
       Action Aapability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS DNS Request Flood
       Action capability";
  }

  identity dns-reply-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS DNS Reply Flood
       Action capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS DNS Reply Flood
       Action capability";
  }

  identity icmp-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS ICMP Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS ICMP Flood Action
       capability";
  }

  identity icmpv6-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS ICMPv6 Flood Action
       capability";
```

```
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS ICMPv6 Flood Action
       capability";
  }

  identity sip-flood-action {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS SIP Flood Action
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS SIP Flood Action
       capability";
  }

  identity detect-mode {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS Detection Mode
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Detection Mode
       capability";
  }

  identity baseline-learning {
    base anti-ddos-capability;
    description
      "Identity for advanced NSF Anti-DDoS Baseline Learning
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF Anti-DDoS Baseline Learning
       capability";
  }

  identity signature-set {
    base ips-capability;
    description
      "Identity for advanced NSF IPS Signature Set capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF IPS Signature Set capability";
  }
```

```
  identity ips-exception-signature {
    base ips-capability;
    description
      "Identity for advanced NSF IPS Exception Signature
       capability";
    reference
      "RFC 8329: Framework for Interface to Network Security
       Functions - Advanced NSF IPS Exception Signature Set
       capability";
  }

  identity voice-id {
    base voip-volte-capability;
    description
      "Identity for advanced NSF VoIP/VoLTE Voice-ID capability";
    reference
      "RFC 3261: SIP: Session Initiation Protocol";
  }

  identity user-agent {
    base voip-volte-capability;
    description
      "Identity for advanced NSF VoIP/VoLTE User Agent capability";
    reference
      "RFC 3261: SIP: Session Initiation Protocol";
  }

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

  identity ike {
    base ipsec-capability;
    description
      "Identity for an IPSec Internet Key Exchange (IKE)
      capability";
  }

  identity ikeless {
    base ipsec-capability;
    description
      "Identity for an IPSec without Internet Key Exchange (IKE)
      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
       draft-ietf-i2nsf-capability-05: Information Model of
       NSFs Capabilities - Capability Information Model Design";

    leaf-list time-capabilities {
      type enumeration {
        enum absolute-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.";
        }
        enum periodic-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.";
        }
      }
      description
        "Time capabilities";
    }

    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 - I2NSF Flow Security Policy Structure
         draft-ietf-i2nsf-capability-05: Information Model of
         NSFs Capabilities - Design Principles and ECA Policy
         Model Overview
         draft-ietf-i2nsf-nsf-monitoring-data-model-03: I2NSF
         NSF Monitoring YANG Data Model - System Alarm and
         System Events";
```

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

      leaf-list system-alarm-capability {
        type identityref {
          base system-alarm-capability;
        }
        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, ICMP, ICMPv6, and payload.";
        reference
          "RFC 791: Internet Protocol - IPv4
           RFC 792: Internet Control Message Protocol - ICMP
           RFC 793: Transmission Control Protocol - TCP
           RFC 768: User Datagram Protocol - UDP
           RFC 8200: Internet Protocol, Version 6 (IPv6)
           Specification - IPv6
           RFC 4443: Internet Control Message Protocol (ICMPv6)
           for the Internet Protocol Version 6 (IPv6) Specification
           - ICMPv6
           RFC 8329: Framework for Interface to Network Security
           Functions - I2NSF Flow Security Policy Structure
           draft-ietf-i2nsf-capability-05: Information Model of
           NSFs Capabilities - Design Principles and ECA Policy
           Model Overview";

        leaf-list ipv4-capability {
          type identityref {
            base ipv4-capability;
          }
```

```
          description
            "IPv4 packet capabilities";
          reference
            "RFC 791: Internet Protocol";
        }

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

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

        leaf-list icmpv6-capability {
          type identityref {
            base icmpv6-capability;
          }
          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-capability;
          }
          description
            "TCP packet capabilities";
          reference
            "RFC 793: Transmission Control Protocol - TCP";
        }
```

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

      container advanced-nsf-capabilities {
        description
          "Advanced Network Security Function (NSF) capabilities,
           such as Anti-Virus, Anti-DDoS, IPS, and VoIP/VoLTE.";
        reference
          "RFC 8329: Framework for Interface to Network Security
           Functions - Advanced NSF capabilities";

        leaf-list anti-virus-capability {
          type identityref {
            base anti-virus-capability;
          }
          description
            "Anti-Virus capabilities";
          reference
            "RFC 8329: Framework for Interface to Network Security
             Functions - Advanced NSF Anti-Virus capabilities";
          }

        leaf-list anti-ddos-capability {
          type identityref {
            base anti-ddos-capability;
          }
          description
            "Anti-DDoS Attack capabilities";
          reference
            "RFC 8329: Framework for Interface to Network Security
             Functions - Advanced NSF Anti-DDoS Attack capabilities";
        }

        leaf-list ips-capability {
          type identityref {
            base ips-capability;
          }
          description
            "Intrusion Prevention System (IPS) capabilities";
          reference
```

```
            "RFC 8329: Framework for Interface to Network Security
             Functions - Advanced NSF IPS capabilities";
        }

        leaf-list url-capability {
          type identityref {
            base url-capability;
          }
          description
            "URL capabilities";
          reference
            "RFC 8329: Framework for Interface to Network Security
             Functions - Advanced NSF URL capabilities";
        }

        leaf-list voip-volte-capability {
          type identityref {
            base voip-volte-capability;
         }
          description
            "VoIP/VoLTE capabilities";
          reference
            "RFC 8329: Framework for Interface to Network Security
             Functions - Advanced NSF VoIP/VoLTE capabilities";
        }
      }

      leaf-list context-capabilities {
        type identityref {
          base context-capability;
        }
        description
          "Security context capabilities";
      }
    }

    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-capability;
        }
        description
```

```
          "Ingress-action capabilities";
      }

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

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

    leaf-list resolution-strategy-capabilities {
      type identityref {
        base resolution-strategy-capability;
      }
      description
        "Resolution strategy capabilities.
         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
         for the same packet and by particular NSF";
      reference
        "draft-ietf-i2nsf-capability-05: Information Model of
         NSFs Capabilities - Resolution strategy capabilities";
    }

    leaf-list default-action-capabilities {
      type identityref {
        base default-action-capability;
      }
      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, alert, or mirror.";
      reference
        "RFC 8329: Framework for Interface to Network Security
         Functions - Ingress and egress actions
         draft-ietf-i2nsf-capability-05: Information Model of
```

```
         NSFs Capabilities - Default action capabilities";
    }

    leaf-list ipsec-method {
      type identityref {
        base ipsec-capability;
      }
      description
        "IPsec method capabilities";
      reference
        "draft-ietf-i2nsf-sdn-ipsec-flow-protection-08:
         Software-Defined Networking (SDN)-based IPsec Flow
         Protection - IPsec methods such as IKE and IKE-less";
    }
  }

  /*
   * 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)";
    }
  }
}

<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][RFC8525].

name: ietf-i2nsf-capability

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

prefix: nsfcap

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.

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.

- ietf-i2nsf-capability: An attacker could alter the security capabilities associated with an NSF whereby disabling or enabling the evasion of security mitigations.

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

- ietf-i2nsf-capability: An attacker could gather the security capability information of any NSF and use this information to evade detection or filtering.

9. References

9.1. Normative References

[draft-ietf-i2nsf-capability]
Xia, L., Strassner, J., Basile, C., and D. Lopez, "Information Model of NSFs Capabilities", draft-ietf-i2nsf-capability-05 (work in progress), April 2019.

[draft-ietf-i2nsf-nsf-monitoring-data-model]
Jeong, J., Chung, C., Hares, S., Xia, L., and H. Birkholz, "I2NSF NSF Monitoring YANG Data Model", draft-ietf-i2nsf-nsf-monitoring-data-model-03 (work in progress), May 2020.

[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-08 (work in progress), June 2020.

[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>.

[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>.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <https://www.rfc-editor.org/info/rfc3444>.

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

[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>.

[RFC768] Postel, J., "User Datagram Protocol", RFC 768, August 1980.

[RFC790] Postel, J., "Assigned Numbers", RFC 790, September 1981.

[RFC791] Postel, J., "Internet Protocol", RFC 791, September 1981.

[RFC792] Postel, J., "Internet Control Message Protocol", RFC 792, September 1981.

[RFC793] Postel, J., "Transmission Control Protocol", RFC 793, September 1981.

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

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

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

[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>.

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

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

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

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

[RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for the Routing Information Base (RIB)", RFC 8431, DOI 10.17487/RFC8431, September 2018, <https://www.rfc-editor.org/info/rfc8431>.

[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>.

9.2. Informative References

[draft-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", draft-ietf-i2nsf-nsf-facing-interface-dm-09 (work in progress), May 2020.

[draft-ietf-i2nsf-registration-interface-dm]
Hyun, S., Jeong, J., Roh, T., Wi, S., and J. Park, "I2NSF Registration Interface YANG Data Model", draft-ietf-i2nsf-registration-interface-dm (work in progress), March 2020.

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 Capabilities of General Firewall

This section shows a configuration example for capabilities registration of general firewall.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>general_firewall</nsf-name>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv4-capability>ipv4-protocol</ipv4-capability>
   <ipv4-capability>exact-ipv4-address</ipv4-capability>
   <ipv4-capability>range-ipv4-address</ipv4-capability>
   <tcp-capability>exact-fourth-layer-port-num</tcp-capability>
   <tcp-capability>range-fourth-layer-port-num</tcp-capability>
  </generic-nsf-capabilities>
 </condition-capabilities>
 <action-capabilities>
  <ingress-action-capability>pass</ingress-action-capability>
  <ingress-action-capability>drop</ingress-action-capability>
  <ingress-action-capability>alert</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>alert</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 4: Configuration XML for Capabilities Registration of General Firewall

Figure 4 shows the configuration XML for capabilities registration of general firewall and its capabilities are as follows.

1. The name of the NSF is general_firewall.

2. The NSF can inspect protocol, exact IPv4 address, and range IPv4 address for IPv4 packets.

3. The NSF can inspect exact port number and range port number for fourth layer packets.

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

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

This section shows a configuration example for capabilities registration of time based firewall.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>time_based_firewall</nsf-name>
 <time-capabilities>absolute-time</time-capabilities>
 <time-capabilities>periodic-time</time-capabilities>
 <condition-capabilities>
  <generic-nsf-capabilities>
   <ipv4-capability>ipv4-protocol</ipv4-capability>
   <ipv4-capability>exact-ipv4-address</ipv4-capability>
   <ipv4-capability>range-ipv4-address</ipv4-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>alert</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>alert</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 5: Configuration XML for Capabilities Registration of Time based Firewall

Figure 5 shows the configuration XML for capabilities registration of time based firewall and 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 protocol, exact IPv4 address, and range IPv4 address for IPv4 packets.

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

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

This section shows a configuration example for capabilities registration of 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-capability>user-defined</url-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>alert</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>alert</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 6: Configuration XML for Capabilities Registration of Web Filter

Figure 6 shows the configuration XML for capabilities registration of web filter and its capabilities are as follows.

1. The name of the NSF is web_filter.

2. The NSF can inspect url for http and https packets.

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

A.4. Example 4: Registration for Capabilities of VoIP/VoLTE Filter

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

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>voip_volte_filter</nsf-name>
 <condition-capabilities>
  <advanced-nsf-capabilities>
   <voip-volte-capability>voice-id</voip-volte-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>alert</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>alert</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 7: Configuration XML for Capabilities Registration of VoIP/ VoLTE Filter

Figure 7 shows the configuration XML for capabilities registration of VoIP/VoLTE filter and its capabilities are as follows.

1. The name of the NSF is voip_volte_filter.

2. The NSF can inspect voice id for VoIP/VoLTE packets.

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

A.5. Example 5: Registration for Capabilities of HTTP and HTTPS Flood Mitigation

This section shows a configuration example for capabilities registration of http and https flood mitigation.

```
<nsf xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
 <nsf-name>http_and_https_flood_mitigation</nsf-name>
 <condition-capabilities>
  <advanced-nsf-capabilities>
   <anti-ddos-capability>http-flood-action</anti-ddos-capability>
   <anti-ddos-capability>https-flood-action</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>alert</ingress-action-capability>
  <egress-action-capability>pass</egress-action-capability>
  <egress-action-capability>drop</egress-action-capability>
  <egress-action-capability>alert</egress-action-capability>
 </action-capabilities>
</nsf>
```

Figure 8: Configuration XML for Capabilities Registration of HTTP and HTTPS Flood Mitigation

Figure 8 shows the configuration XML for capabilities registration of http and https flood mitigation and its capabilities are as follows.

1. The name of the NSF is http_and_https_flood_mitigation.

2. The location of the NSF is 221.159.112.140.

3. The NSF can control the amount of packets for http and https packets.

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

Appendix B. Acknowledgments

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).

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:

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

Authors' Addresses

Susan Hares (editor)
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

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
Fax: +82 31 290 7996
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
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