I2NSF Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 24, 2020

S. Hyun
Myongji University
J. Jeong
T. Roh
S. Wi
Sungkyunkwan University
J. Park
ETRI

January 21, 2020

I2NSF Registration Interface YANG Data Model
draft-ietf-i2nsf-registration-interface-dm-06

Abstract

This document defines an information model and a YANG data model for Registration Interface between Security Controller and Developer's Management System (DMS) in the Interface to Network Security Functions (I2NSF) framework to register Network Security Functions (NSF) of the DMS into the Security Controller. The objective of these information and data models is to support NSF capability registration and query via I2NSF Registration Interface.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

A number of Network Security Functions (NSF) may exist in the Interface to Network Security Functions (I2NSF) framework [RFC8329]. Since each of these NSFs likely has different security capabilities from each other, it is important to register the security capabilities of the NSF into the security controller. In addition, it is required to search NSFs of some required security capabilities on demand. As an example, if additional security capabilities are required to serve some security service request(s) from an I2NSF user, the security controller should be able to request the DMS for NSFs that have the required security capabilities.

This document describes an information model (see Section 5) and a YANG [RFC7950] data model (see Section 6) for the I2NSF Registration Interface [RFC8329] between the security controller and the developer's management system (DMS) to support NSF capability registration and query via the registration interface. It also describes the operations which should be performed by the security controller and the DMS via the Registration Interface using the defined model.

2. Requirements Language

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

3. Terminology

This document uses the following terms defined in [i2nsf-terminology], [capability-dm], [RFC8329], [supa-policy-data-model], and [supa-policy-info-model]

- Network Security Function (NSF): A function that is responsible for a specific treatment of received packets. A Network Security Function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). Sample Network Security Service Functions are as follows: Firewall, Intrusion Prevention/ Detection System (IPS/IDS), Deep Packet Inspection (DPI), Application Visibility and Control (AVC), network virus and malware scanning, sandbox, Data Loss Prevention (DLP), Distributed Denial of Service (DDoS) mitigation and TLS proxy.

- Data Model: 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. [supa-policy-info-model]

- 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. [supa-policy-info-model]

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

4. Objectives

- Registering NSFs to I2NSF framework: Developer's Management System (DMS) in I2NSF framework is typically run by an NSF vendor, and uses Registration Interface to provide NSFs developed by the NSF vendor to Security Controller. DMS registers NSFs and their capabilities to I2NSF framework through Registration Interface. For the registered NSFs, Security Controller maintains a catalog of the capabilities of those NSFs.

- Updating the capabilities of registered NSFs: After an NSF is registered into Security Controller, some modifications on the capability of the NSF may be required later. In this case, DMS uses Registration Interface to update the capability of the NSF, and this update should be reflected in the catalog of NSFs.

- Asking DMS about some required capabilities: In cases that some security capabilities are required to serve the security service request from an I2NSF user, Security Controller searches through the registered NSFs to find ones that can provide the required capabilities. But Security Controller might fail to find any NSFs having the required capabilities among the registered NSFs. In this case, Security Controller needs to request DMS for additional NSF(s) that can provide the required security capabilities via Registration Interface.

5. Information Model

The I2NSF registration interface is used by Security Controller and Developer's Management System (DMS) in I2NSF framework. The following summarizes the operations done through the registration interface:

1) DMS registers NSFs and their capabilities to Security Controller via the registration interface. DMS also uses the registration interface to update the capabilities of the NSFs registered previously.

2) In case that Security Controller fails to find some required capabilities from any registered NSF that can provide , Security Controller queries DMS about NSF(s) having the required capabilities via the registration interface.

Figure 1 shows the information model of the I2NSF registration interface, which consists of two submodels: NSF capability registration and NSF capability query. Each submodel is used for the operations listed above. The remainder of this section will provide in-depth explanations of each submodel.

```
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  |      I2NSF Registration Interface Information Model       |
  |                                                           |
  |         +-+-+-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+-+-+          |
  |         | NSF Capability  |  | NSF Capability  |          |
  |         | Registration    |  | Query           |          |
  |         +-+-+-+-+-+-+-+-+-+-+  +-+-+-+-+-+-+-+-+-+-+          |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: I2NSF Registration Interface Information Model

5.1. NSF Capability Registration

This submodel is used by DMS to register an NSF to Security Controller. Figure 2 shows how this submodel is constructed. The most important part in Figure 2 is the NSF capability, and this specifies the set of capabilities that the NSF to be registered can offer. The NSF Name contains a unique name of this NSF with the specified set of capabilities. When registering the NSF, DMS additionally includes the network access information of the NSF which is required to enable network communications with the NSF.

The following will further explain the NSF capability information and the NSF access information in more detail.

```
                              +-+-+-+-+-+-+-+-+-+
                              | NSF Capability  |
                              | Registration    |
                               +-+-+-+-^-+-+-+-+-+
                                       |
                 +---------------------+--------------------+
                 |                     |                    |
                 |                     |                    |
            +-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+     +-+-+-+-+-+-+-+
            |   NSF     |        | NSF Capability|     | NSF Access  |
            |   Name    |        | Information   |     | Information |
            +-+-+-+-+-+-+        +-+-+-+-+-+-+-+-+     +-+-+-+-+-+-+-+
```

Figure 2: NSF Capability Registration Sub-Model

5.1.1. NSF Capability Information

NSF Capability Information basically describes the security capabilities of an NSF. In Figure 3, we show capability objects of an NSF. Following the information model of NSF capabilities defined in [capability-dm], we share the same I2NSF security capabilities: Time Capabilities, Event Capabilities, Condition Capabilities, Action Capabilities, Resolution Strategy Capabilities, Default Action Capabilities, and IPsec Method [i2nsf-ipsec]. Also, NSF Capability Information additionally contains the performance capabilities of an NSF as shown in Figure 3.

```
                           +-+-+-+-+-+-+-+-+-+
                           | NSF Capability  |
                           |   Information   |
                           +-+-+-+-^-+-+-+-+-+
                                   |
                                   |
            +----------------------+----------------------+
            |                                             |
            |                                             |
    +-+-+-+-+-+-+-+-+-+                             +-+-+-+-+-+-+-+-+-+
    |    I2NSF      |                             |  Performance  |
    | Capabilities  |                             |  Capabilities |
    +-+-+-+-+-+-+-+-+-+                             +-+-+-+-+-+-+-+-+-+
           |
     +--+-----------------+------------------+-----------------+-------+
     |                    |                  |                 |       |
 +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+ |
 |     Time    |   |    Event    |   |  Condition  |   |   Action    | |
 | Capabilities|   | Capabilities|   | Capabilities|   | Capabilities| |
 +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+   +-+-+-+-+-+-+-+ |
                                                                       |
                   +----------------------+--------------------+-------+
                   |                      |                    |
             +-+-+-+-+-+-+-+       +-+-+-+-+-+-+-+       +-+-+-+-+-+-+
             | Resolution  |       |   Default   |       |   IPsec   |
             | Strategy    |       |   Action    |       |   Method  |
             | Capabilities|       | Capabilities|       +-+-+-+-+-+-+
             +-+-+-+-+-+-+-+       +-+-+-+-+-+-+-+
```

Figure 3: NSF Capability Information

5.1.1.1. Performance Capabilities

This information represents the processing capability of an NSF. Assuming that the current workload status of each NSF is being collected through NSF monitoring [i2nsf-monitoring], this capability information of the NSF can be used to determine whether the NSF is in congestion by comparing it with the current workload of the NSF. Moreover, this information can specify an available amount of each type of resource, such as processing power which are available on the NSF. (The registration interface can control the usages and limitations of the created instance and make the appropriate request according to the status.) As illustrated in Figure 4, this information consists of two items: Processing and Bandwidth. Processing information describes the NSF's available processing power. Bandwidth describes the information about available network

amount in two cases, outbound, inbound. These two information can be used for the NSF's instance request.

```
                         +-+-+-+-+-+-+-+-+-+-+
                         |   Performance   |
                         |   Capabilities  |
                         +-+-+-+-+-^-+-+-+-+-+
                                   |
                     +-----------------------------+
                     |                             |
                     |                             |
             +-+-+-+-+-+-+-+-+-+-+           +-+-+-+-+-+-+-+-+
             |  Processing   |           |  Bandwidth  |
             +-+-+-+-+-+-+-+-+-+-+           +-+-+-+-+-+-+-+-+
```

Figure 4: Performance Capability Overview

5.1.2. NSF Access Information

NSF Access Information contains the followings that are required to communicate with an NSF: IPv4 address, IPv6 address, port number, and supported transport protocol(s) (e.g., Virtual Extensible LAN (VXLAN) [RFC 7348], Generic Protocol Extension for VXLAN (VXLAN-GPE) [nvo3-vxlan-gpe], Generic Route Encapsulation (GRE), Ethernet etc.). In this document, NSF Access Information is used to identify a specific NSF instance (i.e. NSF Access Information is the signature(unique identifier) of an NSF instance in the overall system).

5.2. NSF Capability Query

Security Controller may require some additional capabilities to serve the security service request from an I2NSF user, but none of the registered NSFs has the required capabilities. In this case, Security Controller makes a description of the required capabilities by using the NSF capability information sub-model in Section 5.1.1, and sends DMS a query about which NSF(s) can provide these capabilities.

6. Data Model

6.1. YANG Tree Diagram

This section provides the YANG Tree diagram of the I2NSF registration interface.

6.1.1. Definition of Symbols in Tree Diagrams

A simplified graphical representation of the data model is used in this section. The meaning of the symbols used in the following diagrams [RFC8431] 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.

6.1.2. I2NSF Registration Interface

```
          module : ietf-i2nsf-reg-interface
                +--rw nsf-capability-registration
                |  uses nsf-registrations

          rpcs :
                +---x i2nsf-capability-query
                |  uses nsf-capability-query
```

Figure 5: YANG Tree of I2NSF Registration Interface

The I2NSF registration interface is used for the following purposes. Developer's Management System (DMS) registers NSFs and their capabilities into Security Controller via the registration interface. In case that Security Controller fails to find any NSF among the registered NSFs which can provide some required capabilities, Security Controller uses the registration interface to query DMS about NSF(s) having the required capabilities. The following sections describe the YANG data models to support these operations.

6.1.2.1. NSF Capability Registration

This section expands the i2nsf-nsf-registrations in Figure 5.

```
         NSF Capability Registration
          +--rw nsf-registrations
              +--rw nsf-information*  [capability-name]
                 +--rw capability-name                       string
                 +--rw nsf-capability-info
                 |  uses nsf-capability-info
                       +--rw security-capability
                       |  uses ietf-i2nsf-capability
                       +--rw performance-capability
                       |  uses performance-capability
                 +--rw nsf-access-info
                 |  uses nsf-access-info
                       +--rw capability-name
                       +--rw ip
                       +--rw port
```

Figure 6: YANG Tree of NSF Capability Registration Module

When registering an NSF to Security Controller, DMS uses this module to describe what capabilities the NSF can offer. DMS includes the network access information of the NSF which is required to make a network connection with the NSF as well as the capability description of the NSF.

6.1.2.2. NSF Capability Query

This section expands the nsf-capability-query in Figure 5.

```
         I2NSF Capability Query
           +---x nsf-capability-query
               +---w input
               |  +---w query-nsf-capability
               |  |   uses ietf-i2nsf-capability
               +--ro output
                   +--ro nsf-access-info
                   |  uses nsf-access-info
                       +--rw capability-name
                       +--rw ip
                       +--rw port
```

Figure 7: YANG Tree of NSF Capability Query Module

Security Controller may require some additional capabilities to provide the security service requested by an I2NSF user, but none of the registered NSFs has the required capabilities. In this case,

Security Controller makes a description of the required capabilities using this module and then queries DMS about which NSF(s) can provide these capabilities. Use NETCONF RPCs to send a NSF capability query. Input data is query-i2nsf-capability-info and output data is nsf-access-info. In Figure 7, the ietf-i2nsf-capability refers to the module defined in [capability-dm].

6.1.3. NSF Capability Information

This section expands the nsf-capability-info in Figure 6 and Figure 7.

```
      NSF Capability Information
        +--rw nsf-capability-info
          +--rw security-capability
          |  uses ietf-i2nsf-capability
          +--rw performance-capability
          |  uses nsf-performance-capability
```

Figure 8: YANG Tree of I2NSF NSF Capability Information

In Figure 8, the ietf-i2nsf-capability refers to the module defined in [capability-dm]. The performance-capability is used to specify the performance capability of an NSF.

6.1.3.1. NSF Performance Capability

This section expands the nsf-performance-capability in Figure 8.

```
      NSF Performance Capability
        +--rw nsf-performance-capability
         +--rw processing
         |   +--rw processing-average  uint16
         |   +--rw processing-peak     uint16
         +--rw bandwidth
         |   +--rw outbound
         |   |  +--rw outbound-average  uint16
         |   |  +--rw outbound-peak     uint16
         |   +--rw inbound
         |   |  +--rw inbound-average   uint16
         |   |  +--rw inbound-peak      uint16
```

Figure 9: YANG Tree of I2NSF NSF Performance Capability

This module is used to specify the performance capabilities of an NSF when registering or initiating the NSF.

6.1.4. NSF Access Information

This section expands the nsf-access-info in Figure 6.

```
         NSF Access Information
           +--rw nsf-access-info
             +--rw capability-name      string
             +--rw ip      inet:ip-address
             +--rw port    inet:port-number
```

Figure 10: YANG Tree of I2NSF NSF Access Informantion

This module contains the network access information of an NSF that is required to enable network communications with the NSF.

6.2. YANG Data Modules

This section provides YANG modules of the data model for the registration interface between Security Controller and Developer's Management System, as defined in Section 5.

```
    <CODE BEGINS> file "ietf-i2nsf-reg-interface@2020-01-21.yang"

      module ietf-i2nsf-reg-interface {
       yang-version 1.1;

       namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface";

       prefix nsfreg;

       import ietf-inet-types {
        prefix inet;
        reference "RFC 6991";
       }
       import ietf-i2nsf-capability {
        prefix capa;
        reference "draft-ietf-i2nsf-capability-data-model-05";
       }

       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>

         Editor: Sangwon Hyun
```

```
         <mailto:shyun@mju.ac.kr>
         Editor: Jaehoon Paul Jeong
         <mailto:pauljeong@skku.edu>
         Editor: Taekyun Roh
         <mailto:tkroh0198@skku.edu>
         Editor: Sarang Wi
         <mailto:dnl9795@skku.edu>
         Editor: Jung-Soo Park
         <mailto:pjs@etri.re.kr>";

       description
        "This module defines a YANG data model for I2NSF
         registration interface.

         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 XXXX; see
         the RFC itself for full legal notices.";

       revision "2020-01-21" {
        description "Initial revision";
        reference
         "RFC XXXX: I2NSF Registration Interface YANG Data Model";
       }

       container nsf-registrations {
        description
         "Information of an NSF that DMS registers
          to Security Controller";
        list nsf-information {
         key "capability-name";
         description
          "Required information for registration";
         leaf capability-name {
          type string;
          mandatory true;
          description
           "Unique name of this registered NSF";
         }
         container nsf-capability-info {
```

```
          description
           "Capability description of this NSF";
            uses nsf-capability-info;
         }
         container nsf-access-info {
          description
           "Network access information of this NSF";
           uses nsf-access-info;
         }
        }
       }

       grouping nsf-performance-capability {
        description
         "Description of the performance capabilities
          of an NSF";

        container processing {
         description
          "Processing power of an NSF in the unit of GHz (gigahertz)";

         leaf processing-average {
          type uint16;
          description
           "Average processing power";
         }
         leaf processing-peak {
          type uint16;
          description
           "Peak processing power";
         }
        }
        container bandwidth {
         description
          "Network bandwidth available on an NSF
           in the unit of Mbps (megabits per second)";

         container outbound {
          description
           "Outbound network bandwidth";
          leaf outbound-average {
           type uint32;
           units "Mbps";
           description
            "Average outbound bandwidth";
          }
          leaf outbound-peak {
           type uint32;
```

```
           units "Mbps";
           description
            "Peak outbound bandwidth";
          }
         }
         container inbound {
          description
           "Inbound network bandwidth";
          leaf inbound-average {
           type uint32;
           units "Mbps";
           description
            "Average inbound bandwidth";
          }
          leaf inbound-peak {
           type uint32;
           units "Mbps";
           description
            "Peak inbound bandwidth";
          }
         }
        }
       }

       grouping nsf-capability-info {
        description
         "Capability description of an NSF";
        container security-capability {
         description
          "Description of the security capabilities of an NSF";
           uses "capa:nsf-capabilities";
         reference "draft-ietf-i2nsf-capability-data-model-05";
        }
        container performance-capability {
         description
          "Description of the performance capabilities of an NSF";
         uses nsf-performance-capability;
        }
       }

       grouping nsf-access-info {
        description
         "Information required to access an NSF";
        leaf capability-name {
         type string;
         description
           "Unique name of this NSF's capability";
        }
```

```
        leaf ip {
         type inet:ip-address;
         description
          "IPv4/IPv6 address of this NSF";
        }
        leaf port {
         type inet:port-number;
         description
          "Port available on this NSF";
        }
       }

       rpc nsf-capability-query {
        description
         "Description of the capabilities that the
          Security Controller requests to the DMS";
        input {
         container query-nsf-capability {
          description
           "Description of the capabilities to request";
          uses "capa:nsf-capabilities";
          reference
           "draft-ietf-i2nsf-capability-data-model-05";
          }
        }
        output {
         container nsf-access-info {
          description
           "Network access information of an NSF
            with the requested capabilities";
          uses nsf-access-info;
         }
        }
       }
      }


   <CODE ENDS>
```

Figure 11: Registration Interface YANG Data Model

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-reg-interface
      Registrant Contact: The IESG.
      XML: N/A; the requested URI is an XML namespace.
```

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

```
      Name:      ietf-i2nsf-reg-interface
      Namespace: urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface
      Prefix:    nsfreg
      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 secure transport is Secure Shell (SSH) [RFC6242]. The lowest RESTCONF layer is HTTPS, and the required secure transport is TLS [RFC8446].

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

There are a number of data nodes defined in this YANG module that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config) to these data nodes without proper protection can have a negative effect on network operations. These are the subtrees and data nodes and their sensitivity/vulnerability:

- nsf-registrations: The attacker may exploit this to register a compromised or malicious NSF instead of a legitimate NSF to the Security Controller.

- nsf-performance-capability: The attacker may provide incorrect information of the performance capability of any target NSF by illegally modifying this.

- nsf-capability-info: The attacker may provide incorrect information of the security capability of any target NSF by illegally modifying this.

- nsf-access-info: The attacker may provide incorrect network access information of any target NSF by illegally modifying this.

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:

- nsf-registrations: The attacker may try to gather some sensitive information of a registered NSF by sniffing this.

- nsf-performance-capability: The attacker may gather the performance capability information of any target NSF and misuse the information for subsequent attacks.

- nsf-capability-info: The attacker may gather the security capability information of any target NSF and misuse the information for subsequent attacks.

- nsf-access-info: The attacker may gather the network access information of any target NSF and misuse the information for subsequent attacks.

The RPC operation in this YANG module may be considered sensitive or vulnerable in some network environments. It is thus important to control access to this operation. The following is the operation and its sensitivity/vulnerability:

- nsf-capability-query: The attacker may exploit this RPC operation to deteriorate the availability of the DMS and/or gather the information of some interested NSFs from the DMS.

9. References

9.1. Normative References

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

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

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

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

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

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

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

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

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

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

9.2. Informative References

[capability-dm]
Hares, S., Jeong, J., Kim, J., Moskowitz, R., and Q. Lin, "I2NSF Capability YANG Data Model", draft-ietf-i2nsf-capability-data-model-05 (work in progress), July 2019.

[i2nsf-ipsec]
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-07 (work in progress), August 2019.

[i2nsf-monitoring]
Jeong, J., Chung, C., Hares, S., Xia, L., and H. Birkholz, "I2NSF NSF Monitoring YANG Data Model", draft-ietf-i2nsf-nsf-monitoring-data-model-02 (work in progress), November 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-08 (work in progress), July 2019.

[nfv-framework]
"Network Functions Virtualisation (NFV); Architectureal Framework", ETSI GS NFV 002 ETSI GS NFV 002 V1.1.1, October 2013.

[nvo3-vxlan-gpe]
Maino, Ed., F., Kreeger, Ed., L., and U. Elzur, Ed., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-09 (work in progress), December 2019.

[RFC8431] Wang, L., Chen, M., Dass, A., Ananthakrishnan, H., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", RFC 8431, September 2018.

[supa-policy-data-model]
Halpern, J., Strassner, J., and S. van der Meer, "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)", draft-ietf-supa-generic-policy-data-model-04 (work in progress), June 2017.

[supa-policy-info-model]
Strassner, J., Halpern, J., and S. van der 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. XML Example of Registration Interface Data Model

This section describes XML examples of the I2NSF Registration Interface data model under the assumption of registering several types of NSFs and querying NSF capability.

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

This section shows an XML example for registering the capabilities of general firewall.

```
        <nsf-registrations
          xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
          xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
           <nsf-information>
            <capability-name>general_firewall_capability</capability-name>
            <nsf-capability-info>
              <security-capability>
                <condition-capabilities>
                  <generic-nsf-capabilities>
                   <ipv4-capa>capa:ipv4-protocol</ipv4-capa>
                   <ipv4-capa>capa:exact-ipv4-address</ipv4-capa>
                   <ipv4-capa>capa:range-ipv4-address</ipv4-capa>
                   <tcp-capa>capa:exact-tcp-port-num</tcp-capa>
                   <tcp-capa>capa:range-tcp-port-num</tcp-capa>
                  </generic-nsf-capabilities>
               </condition-capabilities>
               <action-capabilities>
                 <ingress-action-capa>capa:pass</ingress-action-capa>
                 <ingress-action-capa>capa:drop</ingress-action-capa>
                 <ingress-action-capa>capa:alert</ingress-action-capa>
                 <egress-action-capa>capa:pass</egress-action-capa>
                 <egress-action-capa>capa:drop</egress-action-capa>
                 <egress-action-capa>capa:alert</egress-action-capa>
               </action-capabilities>
               <ipsec-method>capa:ikeless</ipsec-method>
              </security-capability>
              <performance-capability>
               <processing>
                <processing-average>1000</processing-average>
                <processing-peak>5000</processing-peak>
               </processing>
               <bandwidth>
                <outbound>
                  <outbound-average>1000</outbound-average>
                  <outbound-peak>5000</outbound-peak>
                </outbound>
                <inbound>
```

```
                  <inbound-average>1000</inbound-average>
                  <inbound-peak>5000</inbound-peak>
                </inbound>
               </bandwidth>
             </performance-capability>
           </nsf-capability-info>
           <nsf-access-info>
            <capability-name>general_firewall</capability-name>
            <ip>2001:DB8:8:4::2</ip>
            <port>3000</port>
           </nsf-access-info>
          </nsf-information>
        </nsf-registrations>
```

Figure 12: Configuration XML for Registration of General Firewall

Figure 12 shows the configuration XML for registering the general firewall and its capabilities as follows.

1. The instance 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 tcp packets.

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

5. The NSF can support IPsec not through IKEv2, but through a Security Controller [i2nsf-ipsec].

6. The NSF can have processing power and bandwidth.

7. The location of the NSF is 2001:DB8:8:4::2.

8. The port of the NSF is 3000.

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

This section shows an XML example for registering the capabilities of time-based firewall.

```
      <nsf-registrations
        xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
        xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
```

```
         <nsf-information>
          <capability-name>time_based_firewall_capability</capability-name>
          <nsf-capability-info>
            <security-capability>
              <time-capabilities>absolute-time</time-capabilities>
              <time-capabilities>periodic-time</time-capabilities>
              <condition-capabilities>
              <generic-nsf-capabilities>
                <ipv4-capa>capa:ipv4-protocol</ipv4-capa>
                <ipv4-capa>capa:exact-ipv4-address</ipv4-capa>
                <ipv4-capa>capa:range-ipv4-address</ipv4-capa>
              </generic-nsf-capabilities>
            </condition-capabilities>
            <action-capabilities>
              <ingress-action-capa>capa:pass</ingress-action-capa>
              <ingress-action-capa>capa:drop</ingress-action-capa>
              <ingress-action-capa>capa:alert</ingress-action-capa>
              <egress-action-capa>capa:pass</egress-action-capa>
              <egress-action-capa>capa:drop</egress-action-capa>
              <egress-action-capa>capa:alert</egress-action-capa>
            </action-capabilities>
            <ipsec-method>capa:ike</ipsec-method>
          </security-capability>
          <performance-capability>
            <processing>
              <processing-average>1000</processing-average>
              <processing-peak>5000</processing-peak>
            </processing>
            <bandwidth>
              <outbound>
                <outbound-average>1000</outbound-average>
                <outbound-peak>5000</outbound-peak>
              </outbound>
              <inbound>
                <inbound-average>1000</inbound-average>
                <inbound-peak>5000</inbound-peak>
              </inbound>
            </bandwidth>
          </performance-capability>
        </nsf-capability-info>
        <nsf-access-info>
          <capability-name>time_based_firewall</capability-name>
          <ip>2001:DB8:8:4::3</ip>
          <port>3000</port>
        </nsf-access-info>
      </nsf-information>
    </nsf-registrations>
```

Figure 13: Configuration XML for Registration of Time based Firewall

Figure 13 shows the configuration XML for registering the time-based firewall and its capabilities as follows.

1. The instance name of the NSF is time_based_firewall.

2. The NSF can enforce 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 determine whether the packets are allowed to pass, drop, or alert.

5. The NSF can support IPsec through IKEv2 [i2nsf-ipsec].

6. The NSF can have processing power and bandwidth.

7. The location of the NSF is 2001:DB8:8:4::3.

8. The port of the NSF is 3000.

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

This section shows an XML example for registering the capabilities of web filter.

```
<nsf-registrations
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
  xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
  <nsf-information>
    <capability-name>web_filter</capability-name>
    <nsf-capability-info>
      <security-capability>
        <condition-capabilities>
          <advanced-nsf-capabilities>
            <url-capa>capa:user-defined</url-capa>
          </advanced-nsf-capabilities>
        </condition-capabilities>
        <action-capabilities>
          <ingress-action-capa>capa:pass</ingress-action-capa>
          <ingress-action-capa>capa:drop</ingress-action-capa>
          <ingress-action-capa>capa:alert</ingress-action-capa>
          <egress-action-capa>capa:pass</egress-action-capa>
          <egress-action-capa>capa:drop</egress-action-capa>
          <egress-action-capa>capa:alert</egress-action-capa>
```

```
            </action-capabilities>
            <ipsec-method>capa:ikeless</ipsec-method>
        </security-capability>
        <performance-capability>
          <processing>
            <processing-average>1000</processing-average>
            <processing-peak>5000</processing-peak>
          </processing>
          <bandwidth>
            <outbound>
              <outbound-average>1000</outbound-average>
              <outbound-peak>5000</outbound-peak>
            </outbound>
            <inbound>
              <inbound-average>1000</inbound-average>
              <inbound-peak>5000</inbound-peak>
            </inbound>
          </bandwidth>
        </performance-capability>
      </nsf-capability-info>
      <nsf-access-info>
        <capability-name>web_filter</capability-name>
        <ip>2001:DB8:8:4::4</ip>
        <port>3000</port>
      </nsf-access-info>
    </nsf-information>
  </nsf-registrations>
```

Figure 14: Configuration XML for Registration of Web Filter

Figure 14 shows the configuration XML for registering the web filter, and its capabilities are as follows.

1. The instance name of the NSF is web_filter.

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

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

4. The NSF can support IPsec not through IKEv2, but through a Security Controller [i2nsf-ipsec].

5. The NSF can have processing power and bandwidth.

6. The location of the NSF is 2001:DB8:8:4::4.

7. The port of the NSF is 3000.

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

This section shows an XML example for registering the capabilities of VoIP/VoLTE filter.

```
<nsf-registrations
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
  xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
  <nsf-information>
    <capability-name>voip_volte_filter</capability-name>
    <nsf-capability-info>
      <security-capability>
        <condition-capabilities>
          <advanced-nsf-capabilities>
            <voip-volte-capa>capa:voice-id</voip-volte-capa>
          </advanced-nsf-capabilities>
        </condition-capabilities>
        <action-capabilities>
          <ingress-action-capa>capa:pass</ingress-action-capa>
          <ingress-action-capa>capa:drop</ingress-action-capa>
          <ingress-action-capa>capa:alert</ingress-action-capa>
          <egress-action-capa>capa:pass</egress-action-capa>
          <egress-action-capa>capa:drop</egress-action-capa>
          <egress-action-capa>capa:alert</egress-action-capa>
        </action-capabilities>
        <ipsec-method>capa:ikeless</ipsec-method>
      </security-capability>
      <performance-capability>
      <processing>
        <processing-average>1000</processing-average>
        <processing-peak>5000</processing-peak>
      </processing>
      <bandwidth>
        <outbound>
          <outbound-average>1000</outbound-average>
          <outbound-peak>5000</outbound-peak>
        </outbound>
        <inbound>
          <inbound-average>1000</inbound-average>
          <inbound-peak>5000</inbound-peak>
        </inbound>
      </bandwidth>
    </performance-capability>
  </nsf-capability-info>
  <nsf-access-info>
    <capability-name>voip_volte_filter</capability-name>
```

```
          <ip>2001:DB8:8:4::5</ip>
          <port>3000</port>
        </nsf-access-info>
      </nsf-information>
    </nsf-registrations>
```

Figure 15: Configuration XML for Registration of VoIP/VoLTE Filter

Figure 15 shows the configuration XML for registering VoIP/VoLTE filter, and its capabilities are as follows.

1. The instance name of the NSF is voip_volte_filter.

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

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

4. The NSF can support IPsec not through IKEv2, but through a Security Controller [i2nsf-ipsec].

5. The NSF can have processing power and bandwidth.

6. The location of the NSF is 2001:DB8:8:4::5.

7. The port of the NSF is 3000.

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

This section shows an XML example for registering the capabilities of http and https flood mitigation.

```
<nsf-registrations
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
  xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
  <nsf-information>
    <capability-name>
      http_and_https_flood_mitigation
    </capability-name>
     <nsf-capability-info>
      <security-capability>
         <condition-capabilities>
           <advanced-nsf-capabilities>
            <antiddos-capa>capa:http-flood-action</antiddos-capa>
            <antiddos-capa>capa:https-flood-action</antiddos-capa>
            </advanced-nsf-capabilities>
```

```
                </condition-capabilities>
                <action-capabilities>
                  <ingress-action-capa>capa:pass</ingress-action-capa>
                  <ingress-action-capa>capa:drop</ingress-action-capa>
                  <ingress-action-capa>capa:alert</ingress-action-capa>
                  <egress-action-capa>capa:pass</egress-action-capa>
                  <egress-action-capa>capa:drop</egress-action-capa>
                  <egress-action-capa>capa:alert</egress-action-capa>
                </action-capabilities>
                <ipsec-method>capa:ike</ipsec-method>
           </security-capability>
          <performance-capability>
            <processing>
              <processing-average>1000</processing-average>
              <processing-peak>5000</processing-peak>
            </processing>
            <bandwidth>
              <outbound>
                <outbound-average>1000</outbound-average>
                <outbound-peak>5000</outbound-peak>
              </outbound>
              <inbound>
                <inbound-average>1000</inbound-average>
                <inbound-peak>5000</inbound-peak>
              </inbound>
            </bandwidth>
          </performance-capability>
        </nsf-capability-info>
        <nsf-access-info>
          <capability-name>
            http_and_https_flood_mitigation
          </capability-name>
          <ip>2001:DB8:8:4::6</ip>
          <port>3000</port>
        </nsf-access-info>
      </nsf-information>
    </nsf-registrations>
```

Figure 16: Configuration XML for Registration of of HTTP and HTTPS
Flood Mitigation

Figure 16 shows the configuration XML for registering the http and https flood mitigator, and its capabilities are as follows.

1. The instance name of the NSF is http_and_https_flood_mitigation.

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

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

4. The NSF can support IPsec through IKEv2 [i2nsf-ipsec].

5. The NSF can have processing power and bandwidth.

6. The location of the NSF is 2001:DB8:8:4::6.

7. The port of the NSF is 3000.

A.6. Example 6: Query for Capabilities of Time based Firewall

This section shows an XML example for querying the capabilities of time-based firewall.

```
<rpc message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf-capability-query
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface"
    xmlns:capa="urn:ietf:params:xml:ns:yang:ietf-i2nsf-capability">
     <query-i2nsf-capability-info>
        <time-capabilities>absolute-time</time-capabilities>
        <time-capabilities>periodic-time</time-capabilities>
        <condition-capabilities>
          <generic-nsf-capabilities>
            <ipv4-capa>capa:ipv4-protocol</ipv4-capa>
            <ipv4-capa>capa:exact-ipv4-address</ipv4-capa>
            <ipv4-capa>capa:range-ipv4-address</ipv4-capa>
          </generic-nsf-capabilities>
        </condition-capabilities>
        <action-capabilities>
          <ingress-action-capa>capa:pass</ingress-action-capa>
          <ingress-action-capa>capa:drop</ingress-action-capa>
          <ingress-action-capa>capa:alert</ingress-action-capa>
          <egress-action-capa>capa:pass</egress-action-capa>
          <egress-action-capa>capa:drop</egress-action-capa>
          <egress-action-capa>capa:alert</egress-action-capa>
        </action-capabilities>
        <ipsec-method>capa:ikeless</ipsec-method>
     </query-i2nsf-capability-info>
  </nsf-capability-query>
</rpc>


<rpc-reply message-id="101"
  xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <nsf-access-info
    xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-reg-interface">
    <capability-name>time-based-firewall</capability-name>
    <ip>2001:DB8:8:4::7</ip>
    <port>8080</port>
  </nsf-access-info>
</rpc-reply>
```

Figure 17: Configuration XML for Query of Time-based Firewall

Figure 17 shows the XML configuration for querying the capabilities of the time-based firewall.

Appendix B. NSF Lifecycle Management in NFV Environments

Network Functions Virtualization (NFV) can be used to implement I2NSF framework. In NFV environments, NSFs are deployed as virtual network functions (VNFs). Security Controller can be implemented as an Element Management (EM) of the NFV architecture, and is connected with the VNF Manager (VNFM) via the Ve-Vnfm interface [nfv-framework]. Security Controller can use this interface for the purpose of the lifecycle management of NSFs. If some NSFs need to be instantiated to enforce security policies in the I2NSF framework, Security Controller could request the VNFM to instantiate them through the Ve-Vnfm interface. Or if an NSF, running as a VNF, is not used by any traffic flows for a time period, Security Controller may request deinstantiating it through the interface for efficient resource utilization.

Appendix C. Changes from draft-ietf-i2nsf-registration-interface-dm-05

The following changes have been made from draft-ietf-i2nsf-registration-interface-dm-05:

- This version is revised according to the comments from Reshad Rahman who reviewed this document as a YANG doctor.

Appendix D. 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 E. 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:

- Jinyong Tim Kim (Sungkyunkwan University)

- Chaehong Chung (Sungkyunkwan University)

- Susan Hares (Huawei)

- Diego R. Lopez (Telefonica)

Authors' Addresses

Sangwon Hyun
Department of Computer Engineering
Myongji University
116 Myongji-ro, Cheoin-gu
Yongin, Gyeonggi-do 17058
Republic of Korea

EMail: shyun@mju.ac.kr

Jaehoon Paul Jeong
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

Taekyun Roh
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 290 7222
Fax: +82 31 299 6673
EMail: tkroh0198@skku.edu

Sarang Wi
Department of Electronic, Electrical and Computer Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 290 7222
Fax: +82 31 299 6673
EMail: dnl9795@skku.edu

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon 305-700
Republic of Korea

Phone: +82 42 860 6514
EMail: pjs@etri.re.kr