Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2019

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

July 2, 2018

Registration Interface Information Model
draft-hyun-i2nsf-registration-interface-im-05

Abstract

This document describes an information model for Interface to Network Security Functions (I2NSF) Registration Interface between Security Controller and Developer's Management System (DMS). The information model is required to support NSF instance via the registration interface. This document explains the procedures over I2NSF registration interface for these functionalities. It also describes the detailed information which should be exchanged via I2NSF registration interface.

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 January 3, 2019.

Table of Contents

1. Introduction

A number of virtual network security function instances typically exist in Interface to Network Security Functions (I2NSF) framework [RFC8329]. Since these NSF instances may have different security capabilities, it is important to register the security capabilities of each NSF instance into the security controller after they have been created. In addition, it is required to instantiate NSFs of some required security capabilities on demand. As an example, if additional security capabilities are required to meet the new security requirements that an I2NSF user requests, the security controller should be able to request the DMS to instantiate NSFs that have the required security capabilities.

This document describes the information model which is required for the registration interface between security controller and developer's management system to support registration and instantiation of NSFs. It further describes the procedure based on the information model which should be performed by the security controller and the developer's management system via the registration interface.

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].

3. Terminology

This document uses the terminology described in [i2nsf-terminology][capability-im][RFC8329] [nsf-triggered-steering].

- Network Security Function (NSF): A function that is responsible for 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 [nsf-triggered-steering].

- Advanced Inspection/Action: As like the I2NSF information model for NSF-facing interface [capability-im], Advanced Inspection/ Action means that a security function calls another security function for further inspection based on its own inspection result [nsf-triggered-steering].

- Network Security Function Profile (NSF Profile): NSF Profile specifies the security and performance capability of an NSF instance. Each NSF instance has its own NSF Profile which describes the type of security service it can provide and its performance capabilinty. [nsf-triggered-steering].

4. Objectives

- Registering NSF instances from Developer's Management System: Depending on system's security requirements, it may require some NSFs by default. In this case, DMS creates these default NSF instances without the need of receiving requests from Security

Controller. After creating them, DMS notifies Security Controller of those NSF instances via registration interface.

- Creating an NSF instance to serve another NSF's inspection request: In I2NSF framework, an NSF can trigger another type of NSF(s) for more advanced security inspection of the traffic. In this case, the next NSF is determined by the current NSF's inspection result and client's policy. However, if there is no available NSF instance to serve the former NSF's request, we should create an NSF instance by requesting Developer's Management System (DMS) through registration interface.

- Creating NSF instances required to enforce security policy rules from Client: In I2NSF framework, users decide which security service is necessary in the system. If there is no NSF instances to enforce the client's security policy, then we should also create the required instances by requesting DMS via registration interface.

- Deleting unnecessary NSF instances: In I2NSF framework, users decide which security service is unnecessary in the system. If there is unused NSF instances to enforce the client's security policy, then we should also delete the existing instances by requesting DMS via registration interface.

- Updating an NSF instances: After NSF instance is registered in I2NSF framework, capability of NSF instance can occur added and changed. This situation should be recognized by the security controller of the I2NSF framework. Therefore, if there is updated NSF instances, DMS notifies Security Controller of those NSF instances via registration interface.

5. Information Model

The I2NSF registration interface was only used for registering new NSF instances to Security Controller. In this document, however, we extend its utilization to support on demand NSF instantiation/de-instantiation and describe the information that should be exchanged via the registration interface for the functionality. Moreover, we also define the information model of NSF Profile because, for registration interface, NSF Profile (i.e., capabilities of an NSF) needs to be clarified so that the components of I2NSF framework can exchange the set of capabilities in a standardized manner. This is typically done through the following process:

1) Security Controller first recognizes the set of capabilities (i.e., NSF Profile) or the signature of a specific NSF required or wasted in the current system.

2) Developer’s Management System (DMS) matches the recognized information to an NSF based on the information model definition.

3) Developer’s Management System creates or eliminates NSFs matching with the above information.

4) Security Controller can then add/remove the corresponding NSF instance to/from its list of available NSF instances in the system.

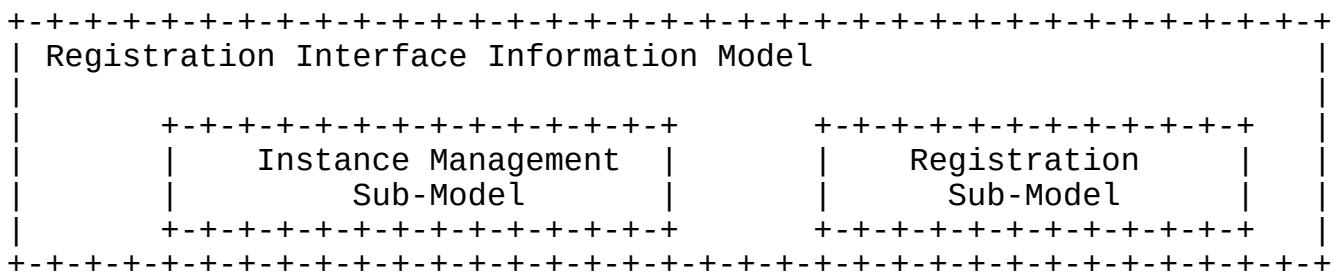

Figure 1: Registration Interface Information Model

As illustrated in Figure 1, the information model for Registration Interface consists of two sub-models: instance management, registration sub-models. The instance management functionality and the registration functionality use NSF Profile to achieve their goals. In this context, NSF Profile is the capability objects that describe and/or prescribe inspection capability an NSF instance can provide.

5.1. NSF Instance Managment Mechanism

For the instance management of NSFs, Security Controller in I2NSF framework requires two types of requests: Instantiation Request and Deinstantiation Request. Security Controller sends the request messages to DMS when required. Once receiving the request, DMS conducts creating/eliminating the corresponding NSF instance and responds Security Controller with the results.

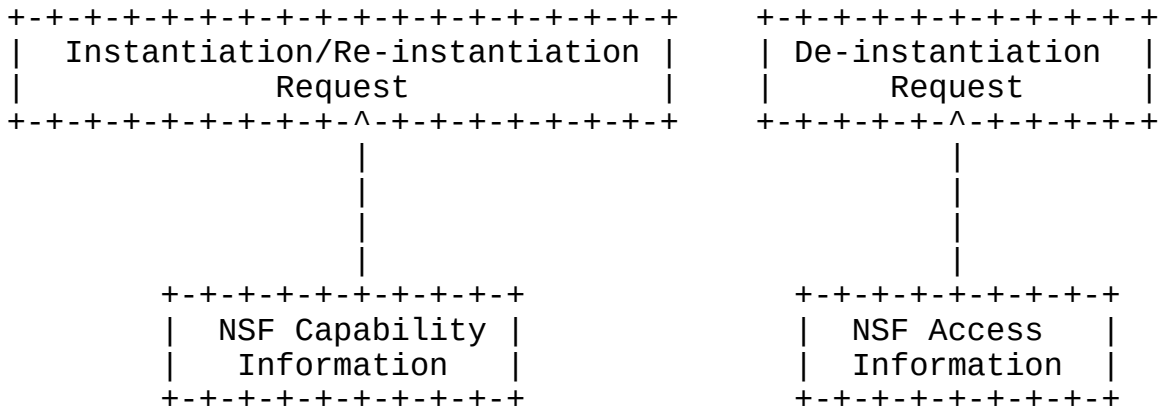

Figure 2: Overview of Instance Management Sub-Model

5.2. NSF Registration Mechanism

In order to register a new NSF instance, DMS should generate a Registration Message to Security Controller. A Registration Message consists of an NSF Profile and an NSF Access Information. The former describes the inspection capability of the new NSF instance and the latter is for enabling network access to the new instance from other components. After this registration process, as explained in [capability-im], the I2NSF capability interface can conduct controlling and monitoring the new registered NSF instance.

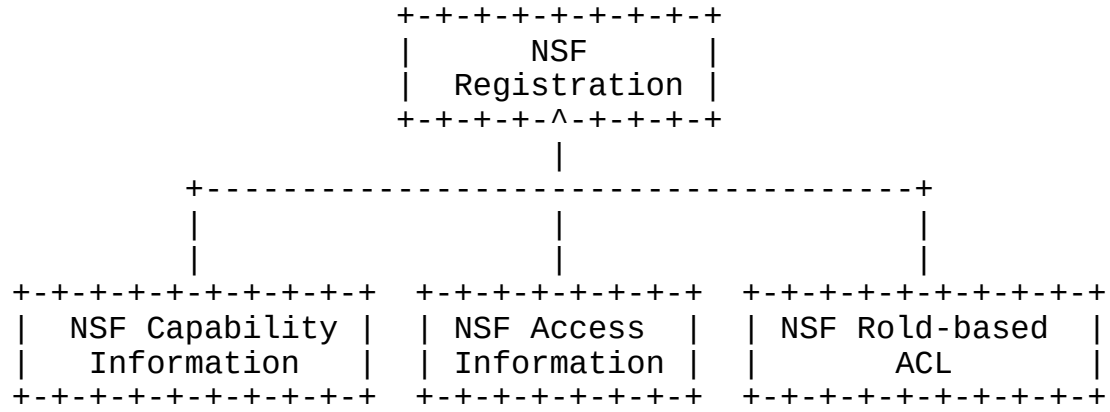

Figure 3: Registration Mechanism Sub-Model Overview

5.3. 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)

[draft-ietf-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).

5.4. NSF Capability Information (Capabilities of an NSF instance)

NSF Profile basically describes the inspection capabilities of an NSF instance. In Figure 4, we show capability objects of an NSF instance. Following the information model of NSF capabilities defiend in [capability-im], we share the same security capabilities: Network-Security Capabilities, Content-Security Capabilities, and Attack Mitigation Capabilities. Also, NSF Profile additionally contains the performance capabilities and role-Based access control list (ACL) as shown in Figure 4.

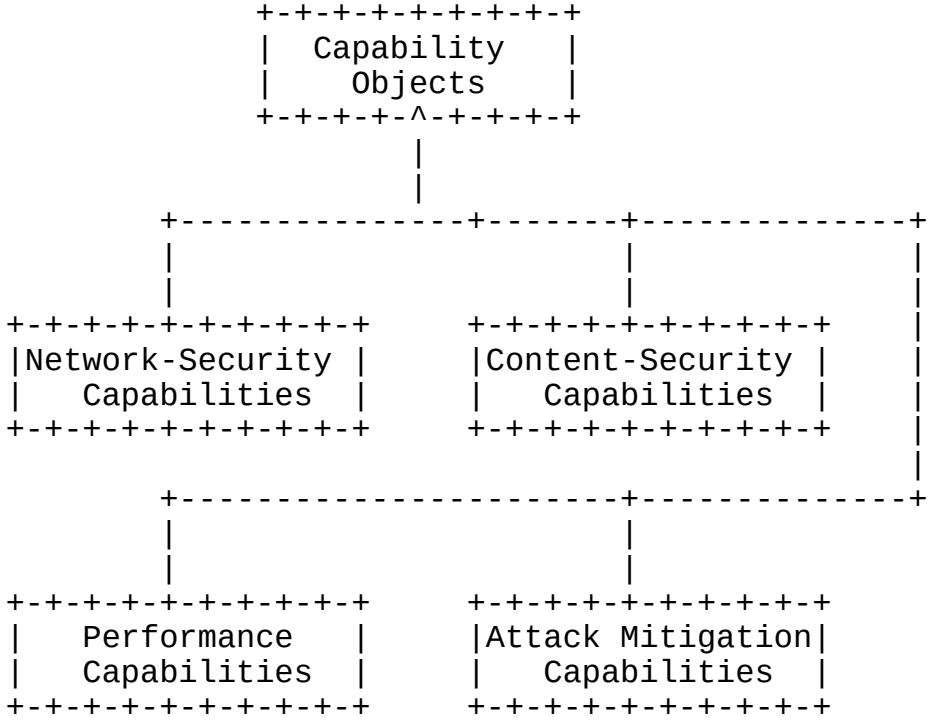

Figure 4: NSF Profile Overview

5.4.1. Performance Capabilities

This information represents the processing capability of an NSF. This information can be used to determine whether the NSF is in congestion by comparing this with the workload that the NSF currently undergoes. Moreover, this information can specify an available amount of each type of resources 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 5, 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. This two information can be used for the NSF's instance request.

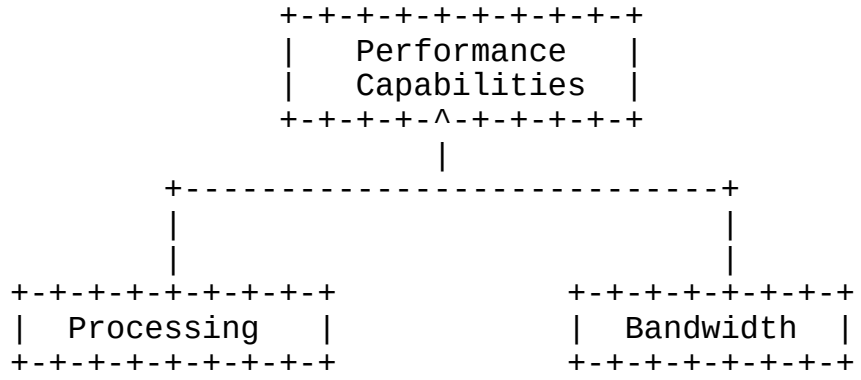

Figure 5: Performance Capability Overview

5.5. Role-based Access Control List

This information specifies access policies of an NSF to determine whether to permit or deny the access of an entity to the NSF based on the role given to the entity. Each NSF is associated with a role-based access control list (ACL) so that it can determine whether to permit or deny the access request from an entity. Figure 6 and Figure 7 show the structure of the role-based ACL, which is composed of role-id, access-type, and permit/deny. The role-id identifies roles of entities (e.g., administrator, developer etc.). The access-type identifies the specific type of access requests such as NSF rule configuration/update and NSF monitoring. Consequently, the role-based ACL in Figure 6 and Figure 7 specifies a set of access-types to be permitted and to be denied by each role-id.

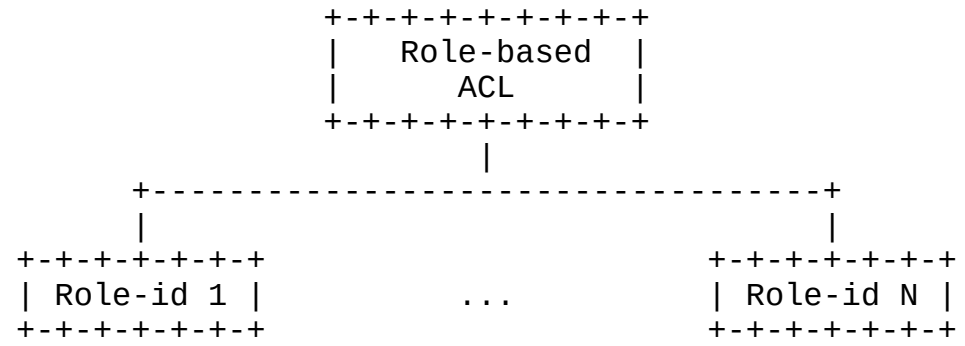

Figure 6: Role-based Access Control List

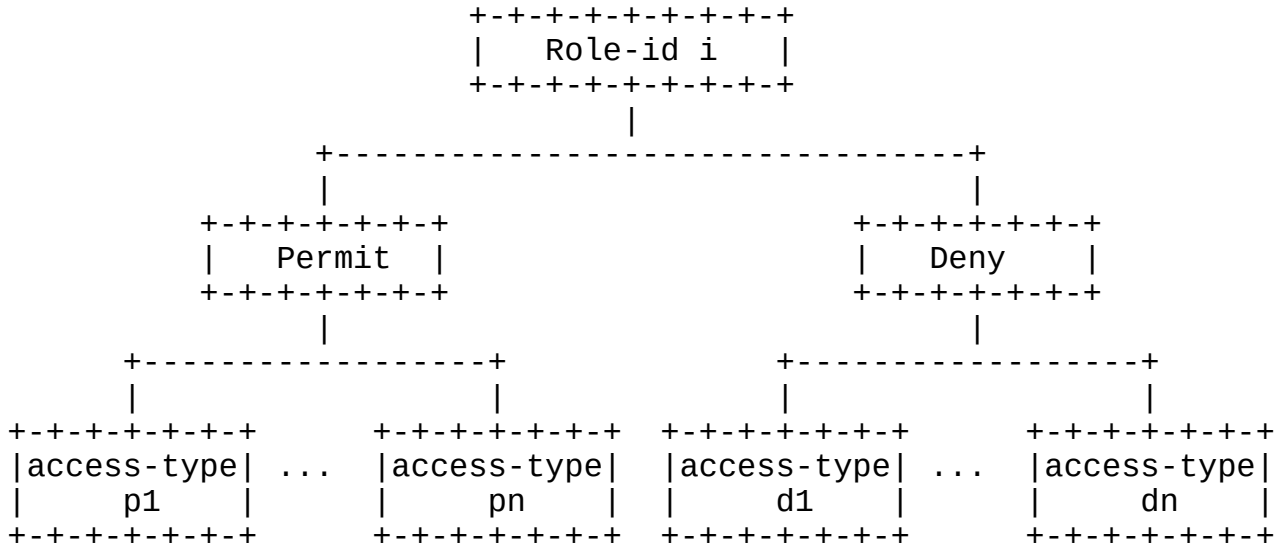

Figure 7: Role-id Subtree

6. Security Considerations

The information model of the registration interface is based on the I2NSF framework without any architectural changes. Thus, this document shares the security considerations of the I2NSF framwork that are specified in [RFC8329] for the purpose of achieving secure communication between components in the proposed architecture.

7. Acknowledgments

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

This document has greatly benefited from inputs by SangUk Woo and YunSuk Yeo.

8. References

8.1. Normative References

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

8.2. Informative References

[capability-im]
Xia, L., Strassner, J., Basile, C., and D. Lopez, "Information Model of NSFs Capabilities", draft-ietf-i2nsf-capability-01 (work in progress), April 2018.

[draft-ietf-nvo3-vxlan-gpe]
Maino, Ed., F., Kreeger, Ed., L., and U. Elzur, Ed., "Generic Protocol Extension for VXLAN", draft-ietf-nvo3-vxlan-gpe-06 (work in progress), April 2018.

[i2nsf-nsf-monitoring]
Hong, D., Jeong, J., Kim, J., Hares, S., Xia, L., and H. Birkholz, "YANG Data Model for Monitoring I2NSF Network Security Functions", draft-hong-i2nsf-nsf-monitoring-data-model-03 (work in progress), March 2018.

[i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-05 (work in progress), January 2018.

[nfv-architecture]
Yang, Hyunsik. and Younghan. Kim, "I2NSF on the NFV Reference Architecture", draft-yang-i2nsf-nfv-architecture-01 (work in progress), March 2018.

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

[nsf-triggered-steering]
Hyun, S., Jeong, J., Park, J., and S. Hares, "Service Function Chaining-Enabled I2NSF Architecture", draft-hyun-i2nsf-nsf-triggered-steering-05 (work in progress), March 2018.

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

Appendix A. Lifecycle Managmenet Mechanism

NFV can be used to virtualize NSFs in I2NSF systems, and DMSs likely perform life-cycle management of NSF instances via Ve-Vnfm interface [nfv-framework] in this environment. The security controller periodically gathers information of NSF instances via the monitoring interface [i2nsf-nsf-monitoring]. This information can be additionally used for life-cycle management of NSF instances, and so the security controller can deliver the information to the DMSs via the registration interface.

Appendix B. Changes from draft-hyun-i2nsf-registration-interface-im-04

The following changes are made from draft-hyun-i2nsf-registration-interface-im-04:

- Section 4 has been revised to discuss about updating an modified NSF instance via registration interface.
- Figures 2, 3 and 4 have been updated.
- Appendix A has been added to discuss about the use of the registration interface related to lifecycle management.
- The references have been updated to reflect the latest documents.

Authors' Addresses

Sangwon Hyun
Department of Computer Engineering
Chosun University
309, Pilmun-daero, Dong-gu
Gwangju, Jeollanam-do 61452
Republic of Korea

EMail: shyun@chosun.ac.kr

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

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

TaeKyun Roh
Department of Software
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
URI: http://imtl.skku.ac.kr/xe/index.php?mid=board_YoKq57

SaRang Wi
Department of Software
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
URI: http://imtl.skku.ac.kr/xe/index.php?mid=board_YoKq57

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