Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 6, 2021

H. Yang
Y. Kim
Soongsil University
J. Jeong
Sungkyunkwan University

November 2, 2020

I2NSF on the NFV Reference Architecture draft-yang-i2nsf-nfv-architecture-06

Abstract

This document describes the integration of Interface to Network Security Functions (I2NSF) Framework into the Network Functions Virtualization (NFV) Reference Model. This document explains how the components and interfaces in the I2NSF Framework can be placed in the NFV reference architecture, and also explains the procedures of the lifecycle management of Network Security Functions (NSFs) according to a user's security policy specification in the I2NSF framework on the NFV system.

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 May 6, 2021.

Table of Contents

1. Introduction

The goal of Interface to Network Security Functions (I2NSF) is to define a set of software interfaces and components for controlling and monitoring aspects of physical and virtual Network Security Functions (NSFs), with which a user can specify high-level security policy. To achieve this goal, the I2NSF framework not only considers physical infrastructure, but also considers a Network Functions Virtualization (NFV) environment since an NSF may be provided by virtualized infrastructure as a Virtual Network Function (VNF). Especially, the I2NSF applicability document [I2NSF-Applicability] describes the applicability of I2NSF to network-based security

services in an NFV environment. Although it explains how I2NSF framework in an NFV environment for security services, it does not explain the procedures of the lifecycle management of NSFs in detail. Thus, this document explains such procedures in the I2NSF framework on the NSF system along with the places of the components of the I2NSF framework in the NFV system.

2. Terminology

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]. This document uses the terminology described in [RFC8329], [I2NSF-Applicability], [ETSI-GS-NFV-003], [Registration-Interface], and [ETSI-GS-IFA-008].

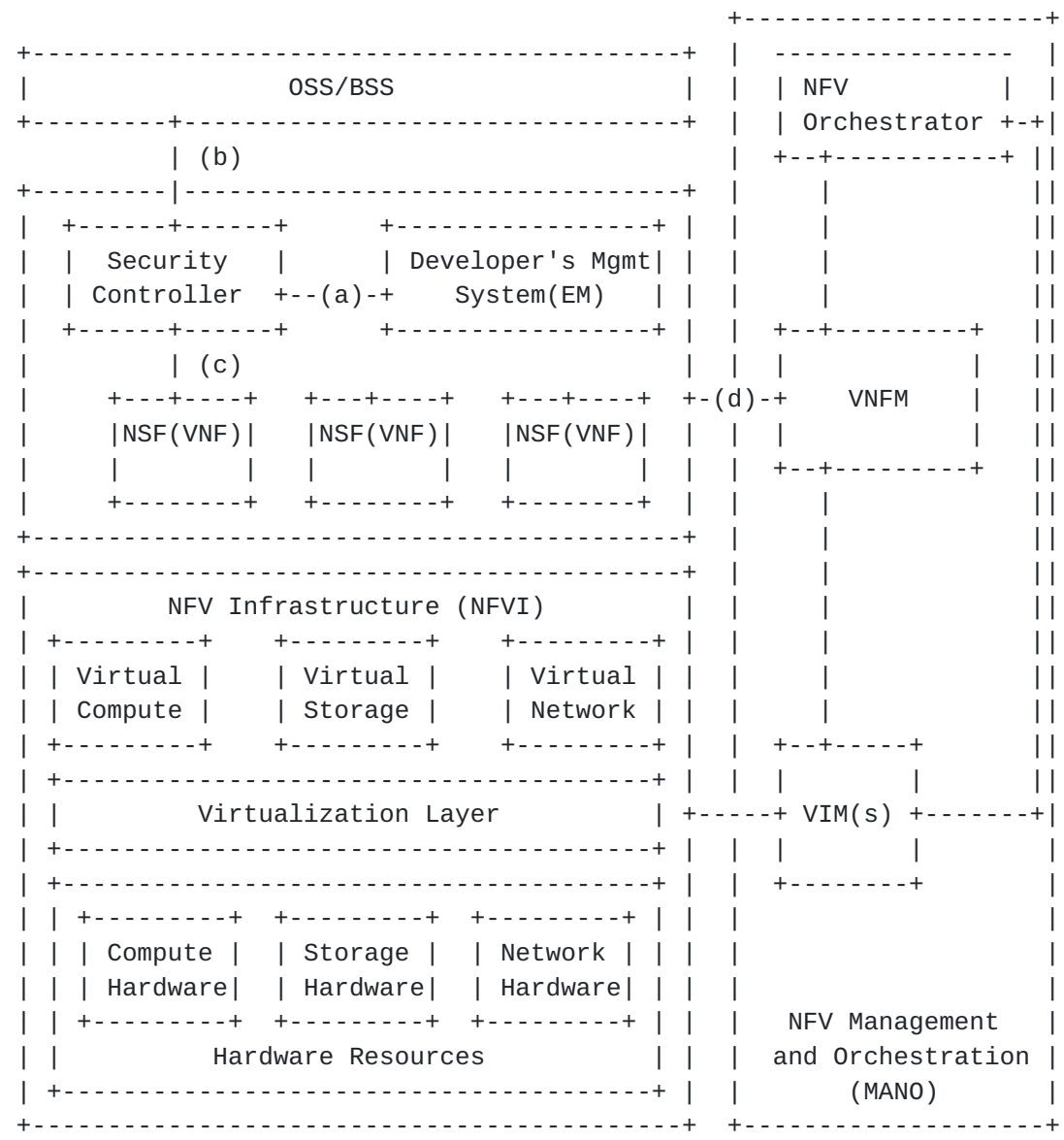
3. I2NSF framework onto the NFV Reference Model

The European Telecommunications Standards Institute (ETSI) defined the components for the basic NFV architecture including the NFV Infrastructure (NFVI), VNF Manager (VNFM), Virtualization Infrastructure Manager (VIM), and NFV Orchestrator (NFVO) [ETSI-GS-NFV-003]. NFVI provides the virtual resources, such as a Virtual Machine (VM) and a Virtual Network, which are used to create, update, and delete VNFs running applications. VNFs are implemented through software virtualization techniques running over the NFVI.

Virtualized Infrastructure Manager (VIM) has a function for controlling and managing the NFVI compute, storage and network nodes, within one operator's infrastructure sub-domain. It also collects and forwards performance measurement data and events.

VNFM manages the VNF lifecycle. When a VNF is created, the VNFM manages the VNF instance in the lifecycle, and the VNFM performs several actions such as software update/modification, monitoring data collection (e.g., fault event in the VNF, and instance termination).

In [RFC8329], the I2NSF framework has four components (i.e., I2NSF User, Security Controller, NSF, and Developer's Management System (DMS)) along with three main interfaces (i.e., Consumer-Facing Interface, NSF-Facing Interface, and Registration Interface). To adopt these components to the NFV reference architecture, each component should be classified based on functionality. According to component functionality, it would correspond to NFV reference architecture components as Figure 1.



(a) = Registration Interface, (b) = Consumer-Facing Interface
(C) = NSF-Facing Interface, (d) = Ve-Vnfm Interface

Figure 1: I2NSF Framework on NFV Reference Architecture

3.1. Network Security Function

A Network Security Function (NSF) is one of the security service functions. In the ETSI reference architecture, a VNF is a network function which provides a specific service. Therefore, an NSF corresponds to a VNF in the NFV reference architecture.

3.2. Security Controller

According to an I2NSF framework, Security Controller has a role to translate an I2NSF User's high-level security policy into a low-level security policy for an NSF. It also collects an NSF's capability information from DMS. Based on the features of the I2NSF framework, Security Controller receives a high-level security policy from an I2NSF user over Consumer-Facing Interface, and after the security policy translation, it can forward the corresponding low-level security policy to an appropriate NSF over NSF-Facing Interface.

In the NFV reference architecture, Element Management (EM) has a role to manage its service function (e.g., firewall, Deep Packet Inspection, DDoS-attack mitigator) and collaborate with the VNF Manager for the lifecycle management (e.g., the instantiation and de-instantiation) of a VNF corresponding to the required security function. This lifecycle management requires the exchange of information regarding the NFVI resources associated with the VNF. EM performs typical management functionality for its own VNFs.

This document proposes that Security Controller can be implemented as an EM that can give a security policy to an NSF, and control the NSF. Note that from the perspective of implementation, it can also be configured as an independent component in the NFV system.

3.3. Developer's Management System

According to the definition of I2NSF Registration Interface, DMS registers its NSF, which can be provided by a specific vendor, into Security Controller along with the capability of the NSF.

In the NFV reference architecture, when a general VNFM creates and manages an NSF, DMS can be used as an EM to manage the specific function of an NSF.

3.4. I2NSF Interfaces

3.4.1. Consumer-Facing Interface

The Consumer-Facing Interface is an interface for communication between I2NSF User and Security Controller. It is used to enable different I2NSF Users in a given I2NSF system to define, manage, and monitor security policies for specific flows within an administrative domain.

In the NFV reference architecture, Operational Support Systems (OSS) and Business Support Systems (BSS) are used to manipulate their applications (e.g., security services) with their policies and rules.

OSS and BSS support a user-domain-specific system for users, such as security enforcement, billing, order, and metering.

Although an interface is not defined between an I2NSF User and a VNF in the NFV reference architecture, Consumer-Facing interface can be deployed for the interaction between an I2NSF user and an VNF, as illustrated in Figure 1.

3.4.2. NSF-Facing Interface

The NSF-Facing Interface is an interface for communication between Security Controller and NSF. It is used to specify and monitor flow-based security policies enforced by one or more NSFs. In the NFV reference architecture, Software Architecture (SWA)-4 Interface is defined. The interface SWA-4 is used by the EM to communicate with a VNF. This management interface is used for the runtime management of the VNF according to Fulfillment, Assurance, Billing, and FCAPS (Fault, Configuration, Accounting, Performance, Security) network management models and frameworks. Therefore, NSF-Facing Interface corresponds to the SWA-4 interface.

3.4.3. Registration Interface

Registration Interface is used to register an NSF from DMS System to Security Controller. An NSF's capabilities can either be pre-configured or retrieved dynamically through the I2NSF Registration Interface. Also, it is to search an appropriate NSF with the required capability that can execute the requested security service.

In the NFV reference architecture, an interface is not defined between EM, the registraion-interface can be deployed like a Figure 1.

3.4.4. Interface for NSF Management

In this model, DMS needs to communicate with VNFM to create an NSF dynamically. This interface is not defined in the I2NSF framework, since it is out of the scope of the I2NSF. However, ETSI defined an interface "Ve-Vnfm" between EM and VNFM [ETSI-GS-IFA-008]. Therefore, as an EM, DMS can use the interface "Ve-Vnfm" to communicate with VNFM.

4. Initial Configuration Procedure in NFV Architecture

The security service procedure in the proposed architecture is as follows. When an I2NSF User requests a security service to Security Controller with a high-level policy, Security Controller translates the high-level policy into the corresponding low-level policy. Then,

it searches the NSF list with capabilities for the requested security service according to the low-level policy. In this step, there are two use cases.

As shown in Figure 2, the first case is the case that when an NSF with the required capability is in active state. The second case is the case that when an NSF with required capability is in inactive state. When the NSF is in active state, Security Controller generates a low-level policy and forwards it to the NSF to set the low-level policy up. On the other hand, when the NSF is in inactive state, Security Controller sends an NSF initiation request message to the DMS via Registration Interface and DMS analyzes the message according to the vendor's configuration [Registration-Interface]. After that, DMS forwards the NSF initiation request message to VNFM via Ve-Vnfm Interface [ETSI-GS-IFA-008]. After the initiation of the NSF, VNFM sends back an NSF initiation response message to Security Controller with an NSF's access information (e.g., IP address, transport-layer protocol, port number, and the NSF's name). With the received NSF access information, Security Controller generates a low-level policy and forwards it to the NSF to set the security policy up.

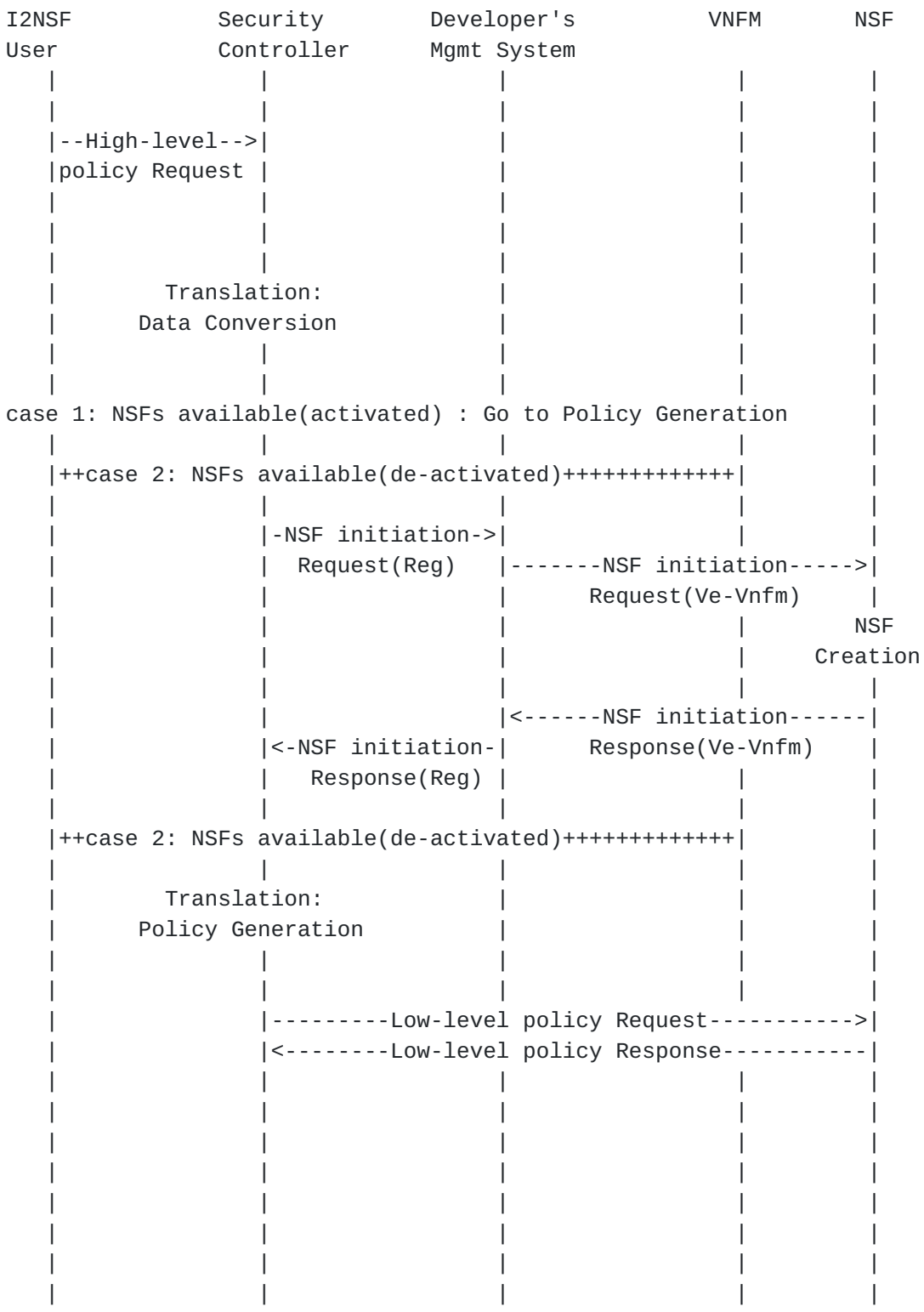


Figure 2: Procedure of I2NSF Framework on NFV for the Case of 'NSF Available'

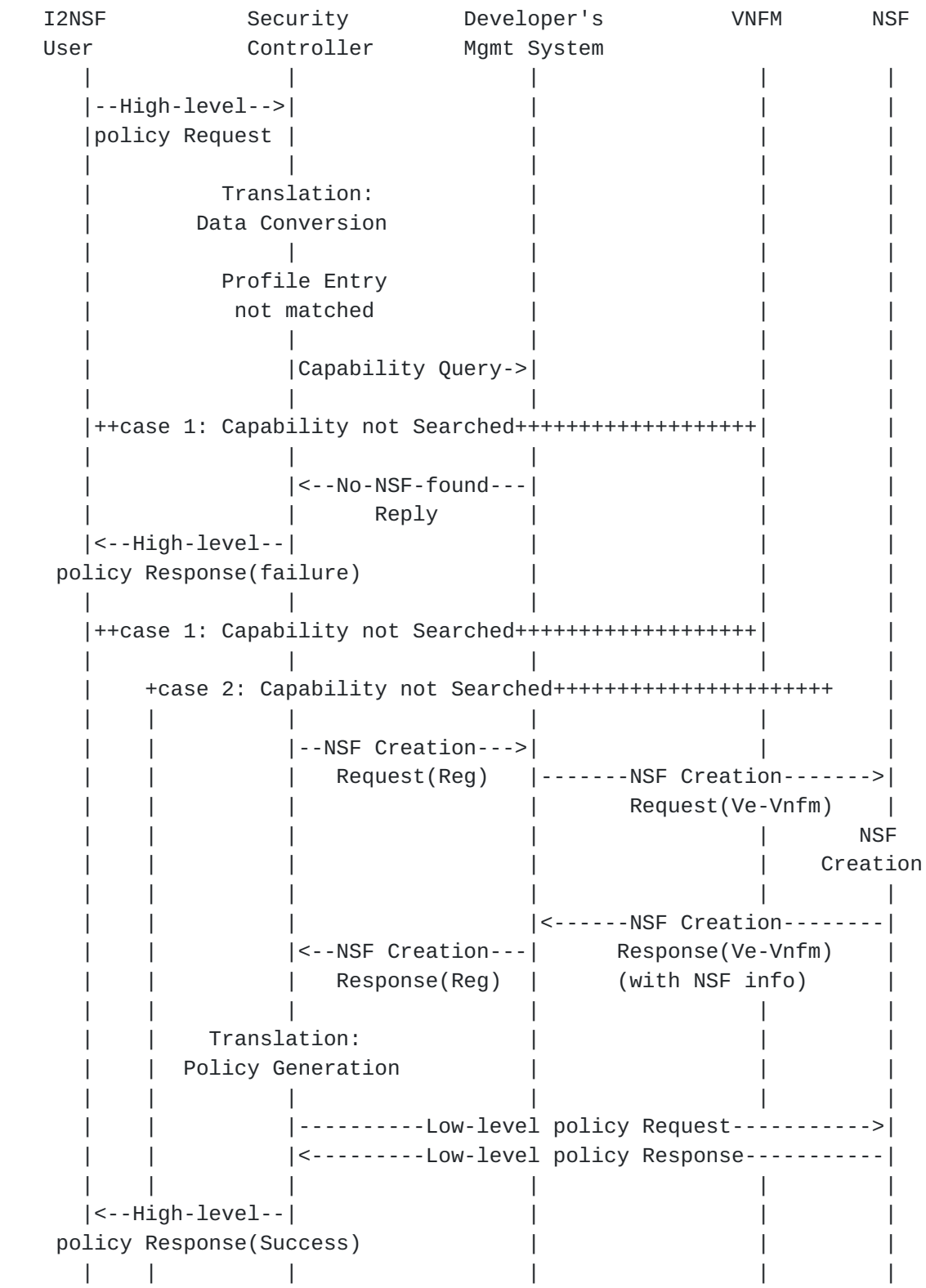

I2NSF User
Security Controller
Developer's Mgmt System
VNFM
NSF
--High-level-->
policy Request
Translation:
Data Conversion
Profile Entry
not matched
Capability Query->
++case 1: Capability not Searched++++++++++++++++++
<--No-NSF-found---
Reply
<--High-level--
policy Response(failure)
++case 1: Capability not Searched++++++++++++++++++
+case 2: Capability not Searched++++++++++++++++++++
--NSF Creation--->
Request(Reg)
-------NSF Creation------->
Request(Ve-Vnfm)
NSF
Creation
<------NSF Creation--------
<--NSF Creation---
Response(Reg)
Response(Ve-Vnfm)
(with NSF info)
Translation:
Policy Generation
----------Low-level policy Request----------->
<---------Low-level policy Response-----------
<--High-level--
policy Response(Success)

Figure 3: Procedure of I2NSF Framework on NFV for the Case of 'No NSF Existing'

However, when the NSF does not exist, The procedure is as follows. As shown in Figure 3, when an NSF does not exist, Security Controller sends a Capability Query message to DMS to search for an NSF with the requested capability. When DMS does not find such an NSF, the procedure is terminated after sending Security Controller a failure notification message, which means that it does not have any NSF with the requested capability. On the other hand, When there exists an NSF corresponding to the requested capability, DMS sends an NSF creation request message to the VNFM. After the creation of the NSF as a VNF, it is registered into DMS. DMS registers the NSF with capability and access information into Security Controller via Registration Interface. The remaining procedure is the same as the previous case.

5. Multi-site Consideration

The previous section described how the I2NSF framework is plugged into the NFV architecture in a single site. From the perspective of NFV, when security functions are deployed, it might be deployed at a single site or multiple sites.

Basically, the I2NSF framework only considers that a single DMS could manage all its NSFs. From the perspective of ETSI reference architecture, when NSFs are deployed at a multi-site environment, a DMS could manage all of the NSFs in such an environment in the same way of a single site. Alternatively, multiple DMSs could manage the NSFs together. The I2NSF framework only considers a single Security Controller that manages all the NSFs in its management domain. This implies that one Security Controller as an EM should be located at the domain.

However, from the perspective of ETSI reference architecture, an EM usually is located at each site and controls a VNF which belongs to that site. The I2NSF framework should consider the placement of Security Controller in a multi-site environment, since there is a conflict between the I2NSF framework and the ETSI NFV reference architecture regarding the placement of Security Controller as an EM.

6. Cloud Native NFV Architecture

To consider the perspective of a cloud native environment, the NFV architecture for I2NSF needs to accommodate an I2NSF framework in a cloud native NFV architecture. ETSI defines NFV reference architecture in a cloud native environment based on [ETSI-NFV-IFA29].

In addition, this standardization is in progress based on [ETSI-NFV-IFA40], [ETSI-NFV-IFA10] and [ETSI-NFV-IFA11].

Based on these standard documents, the functions for resource management and infrastructure management in the cloud native environment are defined, and various types of architectures are proposed. To apply the I2NSF framework to a cloud native environment, the cloud native NFV architecture needs to manage each virtual resource, map it into the corresponding NSF, and register the capability of such an NSF with the I2NSF framework. It also needs to define the interfaces between the I2NSF framework and the cloud native NFV architecture.

7. Use Case of SFC-Enabled I2NSF Framework

A service service in the I2NSF applicability in [I2NSF-Applicability]> requires a forwarding mechanism for cloud-based security services. Especially, a Service Function Chaining (SFC)-enabled I2NSF Architecture in [I2NSF-Applicability] shows a use case that uses SFC as a forwarding mechanism. In addition, it specifies SFC components for I2NSF-based sercurity services (e.g., Classifier and Service Function Forwarder (SFF)) and defines the required functionality of the components. Therefore, the following subsections explain the details of each component and consider how it corresponds to the NFV reference architecture.

7.1. SFC Policy Manager

SFC Policy Manager is a part of Security Controller. It is responsible for interpreting a high-level security policy into a low-level security policy, which is given by I2NSF User. It also handles the delivery of the interpreted policy to SFC Classifier(s) for security function chaining. Moreover, it also generates the information of the security function chaining for the requested security service to SFF(s).

In the NFV reference architecture, Management and Orchestration (MANO) performs similar functions as the SFC Policy Manager. More specifically, the NFV Orchestrator (NFVO) performs on-boarding of a new Network Service (NS), VNF-FG (Forwarding Graph), and VNF Packages. In addition, it manages NS lifecycle (including instantiation, scale-out/in, performance measurement, event correlation, and termination).

Therefore, SFC Policy Manager corresponds to NFVO. In addition, if SFC Policy Manager is a part of Security Controller, this function should be separated from Security Controller, and then be placed in MANO.

7.2. SFC Catalog Manager

SFC Catalog Manager is a part of Security Controller. It is responsible for maintaining the information of every available SF instance such as IP address, transport-layer protocol, port number, service name, and load status. Moreover, it should respond to the queries for available NSF instances from SFC Policy Manager in order to help the generation of a forwarding table entry relevant to a given SFP. It also requests DMS to dynamically instantiate additional SF instances in order to avoid service congestion in an NSF or the elimination of an idle NSF instance to avoid resource waste.

In the NFV reference architecture, SFC Catalog Manager corresponds to an EM since the information related to VNF capability is managed by the EM. Moreover, its functions are similar to Security Controller's as explained before.

7.3. Developer's Management System in SFC-Enabled I2NSF Framework

In the SFC-enabled document, the functions of DMS are extended. For the request message from SFC Catalog Manager, DMS creates additional NSF instances for load balancing and eliminates some of idle NSF instances.

As mentioned above, if DMS manages the NSF's lifecycle indirectly with VNFM, it play a role of a VNFM. VNF lifecycle management includes the instantiation, creation, provisioning, scaling, monitoring, and termination of a VM as a VNF instance. Therefore, DMS corresponds to a specific VNFM.

However, for the scaling performance at a network service level, the role of DMS can be a part of MANO.

7.4. The consideration of SFC-enabled architecture in I2NSF Framework

As mentioned above, when the I2NSF is provided in an NFV environment, various use cases can be provided through SFC technology.

As an NFV point of view, SFC can be provided in two ways. The first way is to configure the SFC between NSF through the individual SDN controller, and the second is to configure the SFC through the network management function in the cloud.

The way to provide traffic steering capabilities may vary depending on the cloud environment, but the Security Controller must request traffic steering to the SDN controller or network function management

via VNFM (using Ve-Vnfm interface). Traffic steering can be provided through physical switches or Virtual Switch.

8. Security Considerations

This document specifies the implementation of the I2NSF framework in the NFV system, so the same security considerations for the I2NSF framework [RFC8329] can be applied to this document.

This document shares all the security issues of NFV that are specified in the "Potential Areas of Concern" section of [ETSI-GS-NFV-SEC-001].

9. IANA Considerations

This document does not require any IANA actions.

10. Acknowledgements

This work was supported by the Ministry of Science and ICT (MSIT) under the ITRC (Information Technology Research Center) support program (IITP-2020-2017-0-01633) supervised by the Institute for Information & Communications Technology Planning & Evaluation (IITP). This work was supported in part by the IITP grant funded by the MSIT (2020-0-00395, Standard Development of Blockchain based Network Management Automation Technology).

11. Informative References

[ETSI-GS-IFA-008]
"Network Functions Virtualisation (NFV);Management and Orchestration;Ve-Vnfm reference point - Interface andInformation Model Specification", October 2016.

[ETSI-GS-NFV-003]
"Network Functions Virtualization (NFV); Architectural Framework", October 2013.

[ETSI-GS-NFV-SEC-001]
"Network Functions Virtualisation (NFV); NFV Security; Problem Statement", October 2014.

[ETSI-NFV-IFA10]
"Network Functions Virtualisation (NFV) Release 3;Management and Orchestration;Functional requirements specification", April 2019.

[ETSI-NFV-IFA11]
"Network Functions Virtualisation (NFV) Release 4;Management and Orchestration;VNF Descriptor and Packaging Specification", September 2020.

[ETSI-NFV-IFA29]
"Network Functions Virtualisation (NFV) Release 3; Architecture; Report on the Enhancements of the NFV architecture towards "Cloud-native" and "PaaS"", November 2019.

[ETSI-NFV-IFA40]
"Network Functions Virtualisation (NFV) Release 4; Management and Orchestration; Requirements for service interfaces and object model for OS container management and orchestration specification", March 2020.

[I2NSF-Applicability]
Jeong, J., Hyun, S., Ahn, T., Hares, S., and D. Lopez, "Applicability of Interfaces to Network Security Functions to Network-Based Security Services", draft-ietf-i2nsf-applicability-18 (work in progress), September 2019.

[Registration-Interface]
Hyun, S., Jeong, J., Roh, T., Wi, S., and J. Park, "I2NSF Registration Interface YANG Data Model", draft-ietf-i2nsf-registration-interface-dm-09 (work in progress), August 2020.

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

[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. Changes from draft-yang-i2nsf-nfv-architecture-05

The following changes have been made from draft-yang-i2nsf-nfv-architecture-05:

- This version includes a cloud native reference architecture to accommodate an I2NSF framework in Section 6.

Authors' Addresses

Hyunsik Yang
School of Electronic Engineering
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul, Seoul 06978
Republic of Korea

Phone: +82 10 9005 7439
EMail: yangun@dcn.ssu.ac.kr

Younghan Kim
School of Electronic Engineering
Soongsil University
369, Sangdo-ro, Dongjak-gu
Seoul, Seoul 06978
Republic of Korea

Phone: +82 10 2691 0904
EMail: younghak@ssu.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