Network Working Group Hyunsik Yang
Internet-Draft Younghan Kim
Intended status: Informational Soongsil University
Expires: April 2019 October 22, 2018

I2NSF on the NFV Reference Architecture
draft-yang-i2nsf-nfv-architecture-03.txt

Abstract

This document describes the adoption of I2NSF Framework onto the Network Functions Virtualization (NFV) Reference Model. In this document, we explain the I2NSF Framework adopted to NFV reference architecture with each corresponding component.

Status of this Memo

Table of Contents

1. Introduction

The goal of I2NSF is to define a set of software interfaces and components for controlling and monitoring aspects of physical and virtual NSFs, enabling clients to specify rules set. To enable I2NSF environment, I2NSF framework not only considers physical infrastructure but also considers the NFV environment since NSF may be provided by virtualized infrastructure as a vnfs. Especially, I2NSF applicability document [i2NSF-applicability] describes the applicability of interface to Network Security Functions(I2NSF) to network-based security services in NFV environment. Although it explains how I2NSF provides security service in NFV environment, it doesn't consider how I2NSF framework adopted onto the NFV reference architecture.

Therefore, we explain the I2NSF framework adopted to NFV reference architecture with each corresponding component.

1.1. 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 RFC-2119 [RFC2119].

This document uses the terminology described in [i2nsf-framework],[i2nsf-terminology], [i2nsf-applicability], [etsi-gs-nfv-003] and [nsf-triggered-steering].

2. 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 VM and virtual network, 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 resources, within one operator's infrastructure sub-domain. It also collects and forwards performance measurements 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 - a fault event in the VNF, and instance termination. According to definition of ETSI, the VNFM is divided into Generic VNFM and Specific VNFM. When the VNFs have their specific methods for provisioning and lifecycle management, a specific VNFM required.

In the I2NSF framework [i2nsf-framework], they defined several components such as NSF, Security controller and Developer's Mgmt System. 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.

```
   +-------------------------------------------+  | +--------------+   |
   |                 OSS/BSS                   |  | | NFV          |   |
   +-------------------------------------------+  | | Orchestrator +-+ |
            Consumer facing interface             | +--+-----------+ | |
   +-------------------------------------------+  |    |             | |
   |  +-----------------------------------+    |  |    |             | |
   |  |      Security Controller(EM)      |    |  |    |             | |
   |  +-----------------+-----------------+    |  | +--+---------+   | |
   |      |     NSF-facing interface  |        |(a)-| Devloper's |   | |
   |  +---+----+    +---+----+    +---+----+   |  | | Mgmt(VNFM) |   | |
   |  |NSF(VNF)|    |NSF(VNF)|    |NSF(VNF)|   |  | +--+---------+   | |
   |  +---+----+    +---+----+    +---+----+   |  |    |             | |
   +------|-------------|-------------|--------+  |    |             | |
          |             |             |           |    |             | |
   +------+-------------+-------------+--------+  |    |             | |
   |         NFV Infrastructure (NFVI)         |  |    |             | |
   | +---------+    +---------+    +---------+ |  |    |             | |
   | | Virtual |    | Virtual |    | Virtual | |  |    |             | |
   | | Compute |    | Storage |    | Network | |  |    |             | |
   | +---------+    +---------+    +---------+ |  | +--+-----+       | |
   | +---------------------------------------+ |  | |        |       | |
   | |         Virtualization Layer          | |--|-| VIM(s) +-------- |
   | +---------------------------------------+ |  | |        |         |
   | +---------------------------------------+ |  | +--------+         |
   | | +---------+  +---------+  +---------+ | |  |                    |
   | | | Compute |  | Storage |  | Network | | |  |                    |
   | | | hardware|  | hardware|  | hardware| | |  |                    |
   | | +---------+  +---------+  +---------+ | |  |                    |
   | |          Hardware resources           | |  |  NFV Management    |
   | +---------------------------------------+ |  | and Orchestration  |
   +-------------------------------------------+  +--------------------+
   (a)= Registration interface
```

Figure 1. I2NSF architecture on NFV reference architecture

2.1. NSF

Network Security Function is one of the security service functions.

In the ETSI reference architecture, VNF(Virtual Network Function)is the network functions which provide specific service.

Therefore, NSF corresponds to the VNF in NFV reference architecture.

2.2. Security Controller

According to I2NSF framework, the security controller has a role which translate policy according to user's request and delivers low level policy to NSFs(manages NSF). It also collects NSF capability from developer's Mgmt System. Based on this information, the security controller forwards policy to NSF.

In the NFV reference architecture, EM has a role that it may be aware of virtualization and collaborate with the VNF Manager to perform those functions that require exchanges of information regarding the NFVI Resources associated with the VNF. EM performs typical management functionality for one or several VNFs.

Therefore, the Security controller corresponds to Element management since it should provide the function which controls NSF and policy. In the case of a distributed security controller model, an interface which is used to communicate between controllers should also be considered.

2.3. Developer's Mgmt System

According to the definition of I2NSF Registration Interface, Developer's Mgmt system registers NSF which can be provided by specific vendor. Developer's Mgmt system also can be one of the vendors too.

In the NFV reference architecture, VNFM manages the VNF lifecycle. It also performs several actions such as software update, monitoring and fault management. Generally, generic VNFM means that only one VNFM handle all of the VNF in the NFV environment. However, if additional VNFMs are required for management of specific VNFs, additional VFNMs can be defined as specific VNFMs.

Therefore, if Developer's Mgmt System manages the NSF lifecycle, it can logically correspond to a specific VNFM.

2.4. Interfaces

2.4.1. Consumer-Facing Interface

The Consumer-Facing Interface is an interface for communication between the User and the Security Controller. It is used to enable

different users of a given I2NSF system to define, manage, and monitor security policies for specific flows within an administrative domain.

In the NFV reference architecture, OSS is Operational Support Systems and BSS stands for Business Support Systems. OSS/BSS support the system for users which relates to infra management such as billing, order and metering.
Although an interface is not defined between User and EM in the NFV reference architecture, Consumer-Facing interface can be deployed between user and EM.

2.4.2. 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 the Fulfillment, Assurance, and Billing and FCAPS(Fault, Configuration, Accounting, Performance, Security) network management models and frameworks.

Therefore, NSF-Facing Interface corresponds to the SWA-4 interface.

2.4.3. Registration Interface

Registration Interface is used to register NSF from Developer's Mgmt System to the security controller. An NSF's capabilities can either be pre-configured or retrieved dynamically through the I2NSF Registration Interface.

Above, this document mentioned that, the Developer's Mgmt System handles the NSF life cycle and this interface corresponds to Ve-Vnfm which is defined in the NFV reference architecture. Ve-Vnfm is defined as IFA008 in ETSI document. IFA008 composed of two interfaces. One is Ve-Vnfm-em, another is Ve-Vnfm-VNF.

If security controller is deployed as an EM, then the registration interface corresponds to Ve-Vnfm-em.

3. I2NSF framework onto the NFV Reference Model(Alternative)

In this chapter, we describe an alternative I2NSF architecture in the NFV environment. As shown in Fig.2, Devloper's Mgmt system corresponds to EM and the security controller can be configured as an independent VNF to perform security controller functions. According to this architecture, all of the interfaces can be adapted directly without additional changes.

```
   +-------------------------------------------+  | ----------------   |
   |                 OSS/BSS                   |  | | NFV          |   |
   +-+-----------------------------------------+  | | Orchestrator +-+ |
     |(C)                                         | +--+-----------+ | |
   +-|-----------------------------------------+  |    |             | |
   | |+-----------------------------------+    |  |    |             | |
   | ||         Devloper's Mgmt (EM)      |    |  |    |             | |
   | |+------------------+----------------+    |  | +--+---------+   | |
   | |   | (a)           |            |        |  | |            |   | |
   | |+---+------+   +---+----+   +---+----+   |  | |    VNFM    |   | |
   | +- Security |   |NSF(VNF)|   |NSF(VNF)|   |  | |            |   | |
   |  |controller|   |        |   |        |   |  | +--+---------+   | |
   |  +---+------+   +---+----+   +---+----+   |  |    |             | |
   +------|-------(b)----|------------|--------+  |    |             | |
   +------+--------------+------------+--------+  |    |             | |
   |         NFV Infrastructure (NFVI)         |  |    |             | |
   | +---------+    +---------+    +---------+ |  |    |             | |
   | | Virtual |    | Virtual |    | Virtual | |  |    |             | |
   | | Compute |    | Storage |    | Network | |  |    |             | |
   | +---------+    +---------+    +---------+ |  | +--+-----+       | |
   | +---------------------------------------+ |  | |        |       | |
   | |         Virtualization Layer          | |--|-| VIM(s) +-------- |
   | +---------------------------------------+ |  | |        |         |
   | +---------------------------------------+ |  | +--------+         |
   | | +---------+  +---------+  +---------+ | |  |                    |
   | | | Compute |  | Storage |  | Network | | |  |                    |
   | | | hardware|  | hardware|  | hardware| | |  |                    |
   | | +---------+  +---------+  +---------+ | |  |                    |
   | |          Hardware resources           | |  |  NFV Management    |
   | +---------------------------------------+ |  | and Orchestration  |
   +-------------------------------------------+  +--------------------+
```

(a)= Registration interface, (b)= NSF-facing interface
(C)= Consumer-facing interface

Figure 2. I2NSF architecture on NFV reference architecture

3.1. Security Controller

According to I2NSF framework, the security controller has a role to translate policy according to user's request and delivers low level policy to NSFs(manages NSF).

Logically the security controller function corresponds to the EM, however, from a deployment scenario the security controller can be configured as an independent VNF to perform security controller functions. In addition, Security controller should be able to communicate with NSFs to manage the NSF.

3.2. Developer's Mgmt System

As defined in I2NSF, the Developer's Mgmt system registers NSF which can be provided by specific vendor i.e. it can create the VNF and manage it.

In the NFV reference architecture, Developer's Mgmt system may correspond to EM. When general VNFM creates and manages the NSF, Devloper's Mgmt system can be used as an EM to manage the specific function for NSF.

3.3. Interfaces

3.3.1. Consumer-Facing Interface

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

In the NFV reference architecture, OSS is Operational Support Systems and BSS stands for Business Support Systems. OSS/BSS support the system for users which relates to infra management such as billing, order and metering.

Although an interface is not defined between User and VNF in the NFV reference architecture, the Consumer-Facing interface can be deployed between user and VNF as illustrated in Fig.2.

3.3.2. 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.
As shown in Fig.2, NSF-Facing Interface is an interface for communication between Security Controller and NSF. In this model, we configured security controller as a VNF.

Therefore, the through the NSF-Facing interface, VNFs should be able to communicate with each other (i.e. security controller to other NSF).

3.3.3. Registration Interface

Registration Interface is used to register NSF from Developer's Mgmt System to the security controller. An NSF's capabilities can either be pre-configured or retrieved dynamically through the I2NSF Registration Interface.

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 the Fulfillment, Assurance, and Billing and FCAPS(Fault, Configuration, Accounting, Performance, Security) network management models and frameworks.

As shown in Fig.2, Registration interface corresponds to the SWA-4. In this model, security controller is configured as a VNF and Devloper's Mgmt system corresponds to the EM. Therefore, SWA-4 can be mapped to both functions logically.

4. Initial configuration procedure in NFV Architecture

This procedure is the initiation procedure of I2NSF in NFV architecture. In the proposed architecture, the procedure is as follows: When one vender wants to provide a security service, the vender registers information which is related to a service such as kinds of service functions and specification of service functions to the Devloper's Mgmt system. The Devloper's Mgmt system forwards it to the VNFM to provide image and Network service specifications. It also registers same information to the Security Controller via registration interface.

```
   USER           Security        Devloper's           VNFM       NSF
                 controller       Mgmt system
    CASE 1            |                 |                |         |
      |-NSF Creation->|                 |                |         |
      |    Request    |                 |                |         |
      |               |                 |                |         |
      |        Check the NSF List       |                |         |
      |          (NSF exist)            |                |         |
      |               |-------NSF initial setting Request--------->|
      |               |<------NSF initial setting Response---------|
      |               |                 |                |         |
      |<-NSF Creation-|                 |                |         |
      |    Response   |                 |                |         |
      |               |                 |                |         |
    CASE 2            |                 |                |         |
      |-NSF Creation->|                 |                |         |
      |    Request    |                 |                |         |
      |               |                 |                |         |
      |        Check the NSF List       |                |         |
      |        (NSF doesn't exist)      |                |         |
      |               |                 |                |         |
      |               |--NSF Creation-->|                |         |
      |               |    Request      |                |         |
      |               |            Translation           |         |
      |               |                 |--NSF Creation->|         |
      |               |                 |                |  NSF    |
      |               |                 |                |Creation |
      |               |                 |<-NSF Creation--|         |
      |               |                 |    Response    |         |
      |               |                 |                |         |
      |               |                 |----NSF ip----->|         |
      |               |                 |    Request     |         |
      |               |                 |                |         |
      |               |                 |<----NSF ip-----|         |
      |               |                 |    Response    |         |
      |               |<--NSF Creation--|                |         |
      |               | Response with IP|                |         |
      |               |                 |                |         |
      |               |-------NSF initial setting Request--------->|
      |               |<------NSF initial setting Response---------|
      |               |                 |                |         |
      |<-NSF Creation-|                 |                |         |
      |    Response   |                 |                |         |
```

Figure 3. Procedure of I2NSF architecture on NFV

When a user requests security service, the security controller checks the NSF list in the controller. If NSF already exists in the same domain, the security controller sends an initial setup request message to the NSF directly. Upon receipt of a response message from the NSF, the security controller then proceeds to forward an initial setup response message to the user.

On the other hand, when a user requests for a security service, and the NSF doesn't exist in the NSF list, the security controller sends an NSF creation request message to Developer's Mgmt system. The Developer's Mgmt system subsequently translates this message and requests NSF creation from the VNFM. After NSF creation, the Developer's Mgmt system requests for the ip address of the newly created NSF for management purposes. The Developer's Mgmt system then reports NSF creation to the security controller with the accompanying ip address. The security controller will update the NSF information and consequently process the request from the user.

5. Multi-site Consideration

In the above section, we described how the I2NSF framework is adopted to NFV architecture in single-site. From a perspective of NFV, when security functions are deployed it might be deployed at a single site or multiple sites.

Basically, I2NSF framework only considers that a single Developer's Mgmt system(VNFM) could manage all the NSFs. As a perspective of ETSI reference architecture, when NSFs are deployed at multi-site environment, Developer's Mgmt system(VNFM) could manage all of the NSFs through a single Developer's Mgmt system. Alternatively, it could manage the NSF through multiple Developer's Mgmt systems. I2NSF framework only considers a single security controller managing all the NSFs in a domain. This implies that one security controller(EM) should be located at one domain.

However, as a perspective of ETSI reference architecture, EM usually located at each site and controls VNF which belongs to that site. The I2NSF framework should consider security controller placement 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(EM).

6. Use case - SFC Enabled I2NSF framework

In the I2NSF WG, some documents mentioned use cases for cloud based security with forwarding mechanism. Especially SFC enabled I2NSF document [nsf-triggered-steering] showed the use case which used SFC as a forwarding mechanism. In addition, it defined additional components and extended functionality of components. Therefore, in the following section, we explain the details of each component and consider how it corresponds to the NFV reference architecture.

6.1. SFC Policy Manager

SFC policy manager is a part of the security controller. It is responsible for interpreting a high level policy into a low-level SFC policy, which is given by I2NSF client. It also handles delivery of the interpreted policy to classifiers for security function chaining. Moreover, it also generates an SF forwarding table and distributes the forwarding information to SFF(s).

In the NFV reference architecture, MANO performs similar functions as the SFC policy manager. More specifically the NFV orchestrator (NFVO) performs on-boarding of new Network Service (NS), VNF-FG(forwarding graph) and VNF Packages. In addition, it manages NS lifecycle (including instantiation, scale-out/in, performance measurements, 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.

6.2. SFC Catalog Manager

SFC catalog manger is a part of the security controller. It is responsible for maintaining the information of every available SF instance such as IP address, supported transport protocol, service name, and load status. Moreover, it should respond to the queries for available SF instances from SFC Policy Manager so as to help to generate a forwarding table entry relevant to a given SFP. It also request Developer's Management System to dynamically instantiate supplementary SF instances to avoid service congestion or the elimination of an existing SF instance to avoid resource waste.

In the NFV reference architecture, SFC catalog manager corresponds to Element management since information which is related to VNF capability is managed by EM. Moreover, this function is similar to security controller as we explained earlier.

6.3. Developer's Mgmt System

In the SFC enabled document, the function of Developer's Mgmt system is extended. Following the request message from SFC catalog manager, it creates additional SF instances and eliminates some of the SF instances.

As mentioned above, if Developer's Mgmt system manages the NSF's lifecycle, it corresponds to a specific VNF Manager. VNF life cycle management includes instantiating, creating, provisioning, scaling, monitoring, and termination of VMs in a VNF instance. Therefore, the Developer's Mgmt system corresponds to a specific VNF Manager.

However, for scaling performed at a network service level, the role of Developer's Mgmt system should extend to the MANOManage and orchestrator).

SFC catalog manger is a part of the security controller. It is responsible for maintaining the information of every available SF instance such as IP address, supported transport protocol, service name, and load status. Moreover, it should respond to the queries for available SF instances from SFC Policy Manager so as to help to generate a forwarding table entry relevant to a given SFP. It also request Developer's Management System to dynamically instantiate supplementary SF instances to avoid service congestion or the elimination of an existing SF instance to avoid resource waste.

In the NFV reference architecture, SFC catalog manager corresponds to Element management since information which is related to VNF capability is managed by EM. Moreover, this function is similar to security controller as we explained earlier.

7. Security Considerations

N/A

8. IANA Considerations

This document has no IANA actions.

9. References

9.1. Normative References

[i2nsf-framework]
Lopez, D., Lopez, E., Dunbar, L.,Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions",draft-ietf-i2nsf-framework-07 (work in progress) ,August 2017.

[i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology",draft-ietf-i2nsf-terminology-04 (work in progress), July 2017.

[etsi-gs-nfv-003]
ETSI NFV ISG, "Network Functions Virtualisation (NFV); Terminology for Main Concepts in NFV", ETSI GS NFV 002 V1.1.1 NFV 002, October 2013, <http://www.etsi.org/deliver/etsi_gs/nfv/001_099/002/01.01.01_60/gs_nfv002v010101p.pdf>

9.2. Informative References

[i2NSF-applicability]

J. Jeong., S. Hyun., T. Ahn., S. Hares., D. Lopez.,' Applicability of Interfaces to Network Security Functions to Network-Based Security Services',draft-ietf-i2nsf-applicability-00(work in progress),October, 2017.

[nsf-triggered-steering]

Hyun, S., Jeong, J., Park, J., and S.Hares, "Service Function Chaining-Enabled I2NSF Architecture",draft-hyun-i2nsf-nsf-triggered-steering-03(work in progress), July 2017.

Authors' Addresses

Hyunsik Yang
Soongsil University
369, Sangdo-ro, Dongjak-gu,
Seoul 156-743, Korea
Email: yangun@dcn.ssu.ac.kr

Younghan Kim
Soongsil University
369, Sangdo-ro, Dongjak-gu,
Seoul 156-743, Korea
Email: younghak@ssu.ac.kr