I2NSF Working Group   J. Jeong
Internet-Draft   Sungkyunkwan University
Intended status: Informational   S. Hyun
Expires: April 25, 2019   Chosun University
                                                               T. Ahn
                                                        Korea Telecom
                                                             S. Hares
                                                               Huawei
                                                             D. Lopez
                                                      Telefonica I+D
                                                     October 22, 2018

Applicability of Interfaces to Network Security Functions to Network-Based Security Services
draft-ietf-i2nsf-applicability-07

Abstract

   This document describes the applicability of Interface to Network
   Security Functions (I2NSF) to network-based security services in
   Network Functions Virtualization (NFV) environments, such as
   firewall, deep packet inspection, or attack mitigation engines.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

   Interface to Network Security Functions (I2NSF) defines a framework
   and interfaces for interacting with Network Security Functions
   (NSFs). The I2NSF framework allows heterogeneous NSFs developed by
   different security solution vendors to be used in the Network
   Functions Virtualization (NFV) environment [ETSI-NFV] by utilizing
   the capabilities of such products and the virtualization of security
   functions in the NFV platform. In the I2NSF framework, each NSF
   initially registers the profile of its own capabilities into the
   system in order for themselves to be available in the system. In
   addition, the Security Controller is validated by the I2NSF Client
   (also called I2NSF User) that the user is employing, so that the user
   can request security services through the Security Controller.

   This document illustrates the applicability of the I2NSF framework
   with four different scenarios: (i) the enforcement of time-dependent
   web access control; (ii) the application of I2NSF to a Service
   Function Chaining (SFC) environment [RFC7665]; (iii) the integration
   of the I2NSF framework with Software-Defined Networking (SDN)
   [RFC7149] to provide different security functionality such as
   firewalls [opsawg-firewalls], Deep Packet Inspection (DPI), and
   Distributed Denial of Service (DDoS) attack mitigation; (iv) the use
   of NFV as supporting technology. The implementation of I2NSF in
   these scenarios has allowed us to verify the applicability and
   effectiveness of the I2NSF framework for a variety of use cases.

2. Terminology

   This document uses the terminology described in [RFC7149],
   [ITU-T.Y.3300], [ONF-OpenFlow], [ONF-SDN-Architecture],
   [ITU-T.X.1252], [ITU-T.X.800], [RFC8329], [i2nsf-terminology],
   [consumer-facing-inf-im], [consumer-facing-inf-dm],
   [i2nsf-nsf-cap-im], [nsf-facing-inf-dm], [registration-inf-dm], and
   [nsf-triggered-steering]. In addition, the following terms are
   defined below:

   o Software-Defined Networking (SDN): A set of techniques that
      enables to directly program, orchestrate, control, and manage
      network resources, which facilitates the design, delivery and
      operation of network services in a dynamic and scalable manner
      [ITU-T.Y.3300].

   o Firewall: A service function at the junction of two network
      segments that inspects every packet that attempts to cross the
      boundary. It also rejects any packet that does not satisfy
      certain criteria for, for example, disallowed port numbers or IP
      addresses.

o Centralized Firewall System: A centralized firewall that can
   establish and distribute policy rules into network resources for
   efficient firewall management.

o Centralized VoIP Security System: A centralized security system
   that handles the security functions required for VoIP and VoLTE
   services.

o Centralized DDoS-attack Mitigation System: A centralized mitigator
   that can establish and distribute access control policy rules into
   network resources for efficient DDoS-attack mitigation.

```
     +------------+
     | I2NSF User |
     +------------+
           ^
           | Consumer-Facing Interface
           v
+-------------------+      Registration    +-----------------------+
|Security Controller|<-------------------->|Developer's Mgmt System|
+-------------------+       Interface      +-----------------------+
           ^
           | NSF-Facing Interface
           v
    +----------------+ +---------------+   +-----------------------+
    |     NSF-1      |-|    NSF-2      |...|         NSF-n         |
    |   (Firewall)   | | (Web Filter) |   |(DDoS-Attack Mitigator)|
    +----------------+ +---------------+   +-----------------------+
```
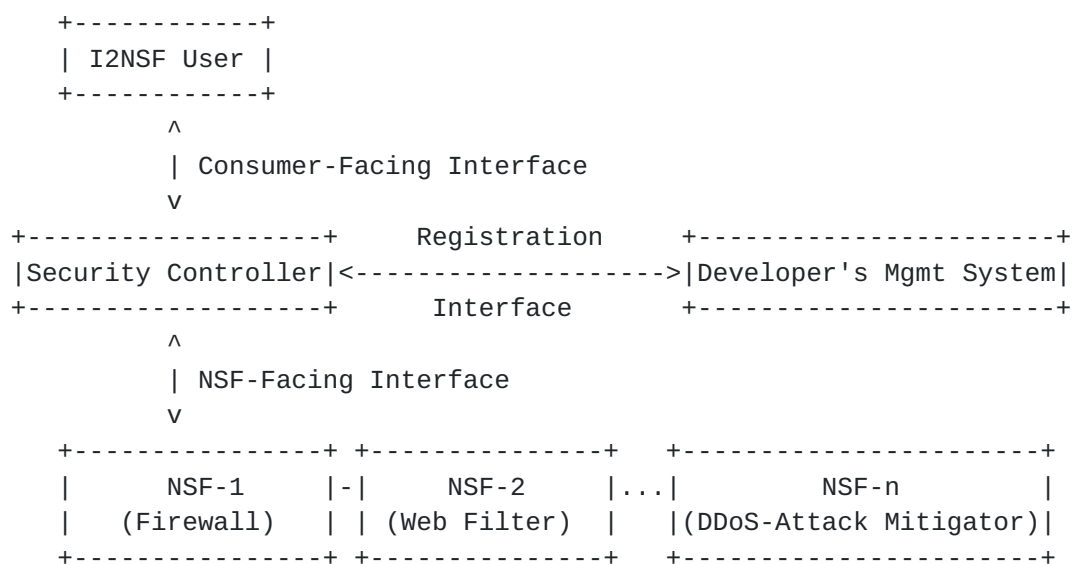
                      Figure 1: I2NSF Framework

3. I2NSF Framework

This section summarizes the I2NSF framework as defined in [RFC8329].
As shown in Figure 1, an I2NSF User can use security functions by
delivering high-level security policies, which specify security
requirements that the I2NSF user wants to enforce, to the Security
Controller via the Consumer-Facing Interface
[consumer-facing-inf-im][consumer-facing-inf-dm].

The Security Controller receives and analyzes the high-level security
policies from an I2NSF User, and identifies what types of security
capabilities are required to meet these high-level security policies.
The Security Controller then identifies NSFs that have the required
security capabilities, and generates low-level security policies for
each of the NSFs so that the high-level security policies are
eventually enforced by those NSFs [policy-translation]. Finally, the

Security Controller sends the generated low-level security policies to the NSFs [i2nsf-nsf-cap-im][nsf-facing-inf-dm].

The Security Controller requests NSFs to perform low-level security services via the NSF-Facing Interface. The developers (or vendors) inform the Security Controller of the capabilities of the NSFs through the I2NSF Registration Interface [registration-inf-dm] for registering (or deregistering) the corresponding NSFs.

The Consumer-Facing Interface between an I2NSF User and the Security Controller can be implemented using, for example, RESTCONF [RFC8040]. Data models specified by YANG [RFC6020] describe high-level security policies to be specified by an I2NSF User. The data model defined in [consumer-facing-inf-dm] can be used for the I2NSF Consumer-Facing Interface.

The NSF-Facing Interface between the Security Controller and NSFs can be implemented using NETCONF [RFC6241]. YANG data models describe low-level security policies for the sake of NSFs, which are translated from the high-level security policies by the Security Controller. The data model defined in [nsf-facing-inf-dm] can be used for the I2NSF NSF-Facing Interface.

The Registration Interface between the Security Controller and the Developer's Management System can be implemented by RESTCONF [RFC8040]. The data model defined in [registration-inf-dm] can be used for the I2NSF Registration Interface.

Also, the I2NSF framework can enforce multiple chained NSFs for the low-level security policies by means of SFC techniques for the I2NSF architecture described in [nsf-triggered-steering].

The following sections describe different security service scenarios illustrating the applicability of the I2NSF framework.

4. Time-dependent Web Access Control Service

This service scenario assumes that an enterprise network administrator wants to control the staff members' access to a particular Interner service (e.g., Example.com) during business hours. The following is an example high-level security policy rule that the administrator requests: Block the staff members' access to Example.com from 9 AM to 6 PM. The administrator sends this high-level security policy to the Security Controller, then the Security Controller identifies required security capabilities, e.g., IP address and port number inspection capabilities and URL inspection capability. In this scenario, it is assumed that the IP address and port number inspection capabilities are required to check whether a

received packet is an HTTP packet from a staff member. The URL
inspection capability is required to check whether the target URL of
a received packet is in the Example.com domain or not.

The Security Controller maintains the security capabilities of each
NSF running in the I2NSF system, which have been reported by the
Developer's Management System via the Registation interface. Based
on this information, the Security Controller identifies NSFs that can
perform the IP address and port number inspection and URL inspection
[policy-translation]. In this scenario, it is assumed that an NSF of
firewall has the IP address and port number inspection capabilities
and an NSF of web filter has URL inspection capability.

The Security Controller generates low-level security rules for the
NSFs to perform IP address and port number inspection, URL
inspection, and time checking. Specifically, the Security Controller
may interoperate with an access control server in the enterprise
network in order to retrieve the information (e.g., IP address in
use, company identifier (ID), and role) of each employee that is
currently using the network. Based on the retrieved information, the
Security Controller generates low-level security rules to check
whether the source IP address of a received packet matches any one
being used by a staff member. In addition, the low-level security
rules should be able to determine that a received packet is of HTTP
protocol. The low-level security rules for web filter checks that
the target URL field of a received packet is equal to Example.com.
Finally, the Security Controller sends the low-level security rules
of the IP address and port number inspection to the NSF of firewall
and the low-level rules for URL inspection to the NSF of web filter.

The following describes how the time-dependent web access control
service is enforced by the NSFs of firewall and web filter.

1. A staff member tries to access Example.com during business hours,
    e.g., 10 AM.

2. The packet is forwarded from the staff member's device to the
    firewall, and the firewall checks the source IP address and port
    number. Now the firewall identifies the received packet is an
    HTTP packet from the staff member.

3. The firewall triggers the web filter to further inspect the
    packet, and the packet is forwarded from the firewall to the web
    filter. SFC technology can be utilized to support such packet
    forwarding in the I2NSF framework [nsf-triggered-steering].

4. The web filter checks the target URL field of the received
    packet, and realizes the packet is toward Example.com. The web

filter then checks that the current time is in business hours.
If so, the web filter drops the packet, and consequently the
staff member's access to Example.com during business hours is
blocked.

```
      +------------+
      | I2NSF User |
      +------------+
            ^
            | Consumer-Facing Interface
            v
  +------------------+      Registration     +----------------------+
  |Security Controller|<-------------------->|Developer's Mgmt System|
  +------------------+       Interface       +----------------------+
      ^         ^
      |         | NSF-Facing Interface
      |         |------------------------
      |                              |
      | NSF-Facing Interface         |
  +------v-----------+        +------v-------+
  |  +-----------+  |    ------>|    NSF-1      |
  |  |Classifier |  |    |      |  (Firewall)  |
  |  +-----------+  |    |      +--------------+
  |    +-----+     |<-----|      +--------------+
  |    | SFF |     |    |----->|    NSF-2      |
  |    +-----+     |    |      |   (DPI)      |
  +----------------+    |      +--------------+
                        |            .
                        |            .
                        |            .
                        |      +----------------------+
                        ------>|      NSF-n            |
                               |(DDoS-Attack Mitigator)|
                               +----------------------+
```
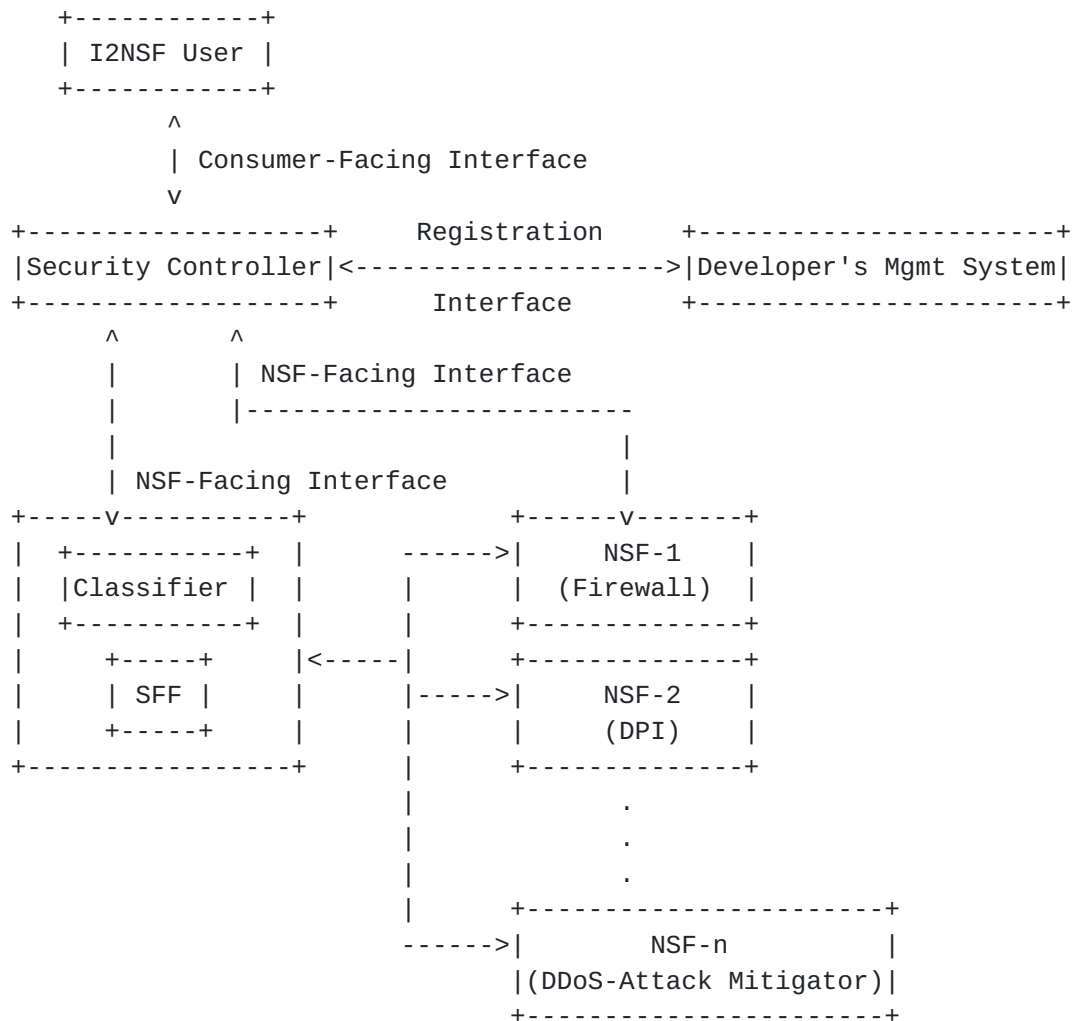
Figure 2: An I2NSF Framework with SFC

5. I2NSF Framework with SFC

In the I2NSF architecture, an NSF can trigger an advanced security
action (e.g., DPI or DDoS attack mitigation) on a packet based on the
result of its own security inspection of the packet. For example, a
firewall triggers further inspection of a suspicious packet with DPI.
For this advanced security action to be fulfilled, the suspicious
packet should be forwarded from the current NSF to the successor NSF.
SFC [RFC7665] is a technology that enables this advanced security
action by steering a packet with multiple service functions (e.g.,

NSFs), and this technology can be utilized by the I2NSF architecture
to support the advanced security action.

Figure 2 shows an I2NSF framework with the support of SFC. As shown
in the figure, SFC generally requires classifiers and service
function forwarders (SFFs); classifiers are responsible for
determining which service function path (SFP) (i.e., an ordered
sequence of service functions) a given packet should pass through,
according to pre-configured classification rules, and SFFs perform
forwarding the given packet to the next service function (e.g., NSF)
on the SFP of the packet by referring to their forwarding tables. In
the I2NSF architecture with SFC, the Security Controller can take
responsibilities of generating classification rules for classifiers
and forwarding tables for SFFs. By analyzing high-level security
policies from I2NSF users, the Security Controller can construct SFPs
that are required to meet the high-level security policies, generates
classification rules of the SFPs, and then configures classifiers
with the classification rules over NSF-Facing Interface so that
relevant traffic packets can follow the SFPs. Also, based on the
global view of NSF instances available in the system, the Security
Controller constructs forwarding tables, which are required for SFFs
to forward a given packet to the next NSF over the SFP, and
configures SFFs with those forwarding tables over NSF-Facing
Interface.

To trigger an advanced security action in the I2NSF architecture, the
current NSF appends a metadata describing the security capability
required for the advanced action to the suspicious packet and sends
the packet to the classifier. Based on the metadata information, the
classifier searches an SFP which includes an NSF with the required
security capability, changes the SFP-related information (e.g.,
service path identifier and service index [RFC8300]) of the packet
with the new SFP that has been found, and then forwards the packet to
the SFF. When receiving the packet, the SFF checks the SFP-related
information such as the service path identifier and service index
contained in the packet and forwards the packet to the next NSF on
the SFP of the packet, according to its forwarding table.

6. I2NSF Framework with SDN

This section describes an I2NSF framework with SDN for I2NSF
applicability and use cases, such as firewall, deep packet
inspection, and DDoS-attack mitigation functions. SDN enables some
packet filtering rules to be enforced in network forwarding elements
(e.g., switch) by controlling their packet forwarding rules. By
taking advantage of this capability of SDN, it is possible to
optimize the process of security service enforcement in the I2NSF
system.

```
        +------------+
        | I2NSF User |
        +------------+
              ^
              | Consumer-Facing Interface
              v
   +------------------+      Registration     +----------------------+
   |Security Controller|<-------------------->|Developer's Mgmt System|
   +------------------+        Interface      +----------------------+
     ^      ^
     |      | NSF-Facing Interface
     |      v
     | +----------------+ +---------------+   +----------------------+
     | |     NSF-1      |-|     NSF-2      |...|        NSF-n          |
     | |   (Firewall)   | |     (DPI)      |   |(DDoS-Attack Mitigator)|
     | +----------------+ +---------------+   +----------------------+
     |
     |
     |                                           SDN Network
   +--|-----------------------------------------------------------------+
   |  V NSF-Facing Interface                                            |
   |  +----------------+                                                |
   |  | SDN Controller |                                                |
   |  +----------------+                                                |
   |          ^                                                         |
   |          | SDN Southbound Interface                               |
   |          v                                                         |
   |      +--------+ +------------+ +--------+      +--------+           |
   |      |Switch-1|-|  Switch-2  |-|Switch-3|.......|Switch-m|          |
   |      |        | |(Classifier)| | (SFF)  |      |        |          |
   |      +--------+ +------------+ +--------+      +--------+           |
   +--------------------------------------------------------------------+
```
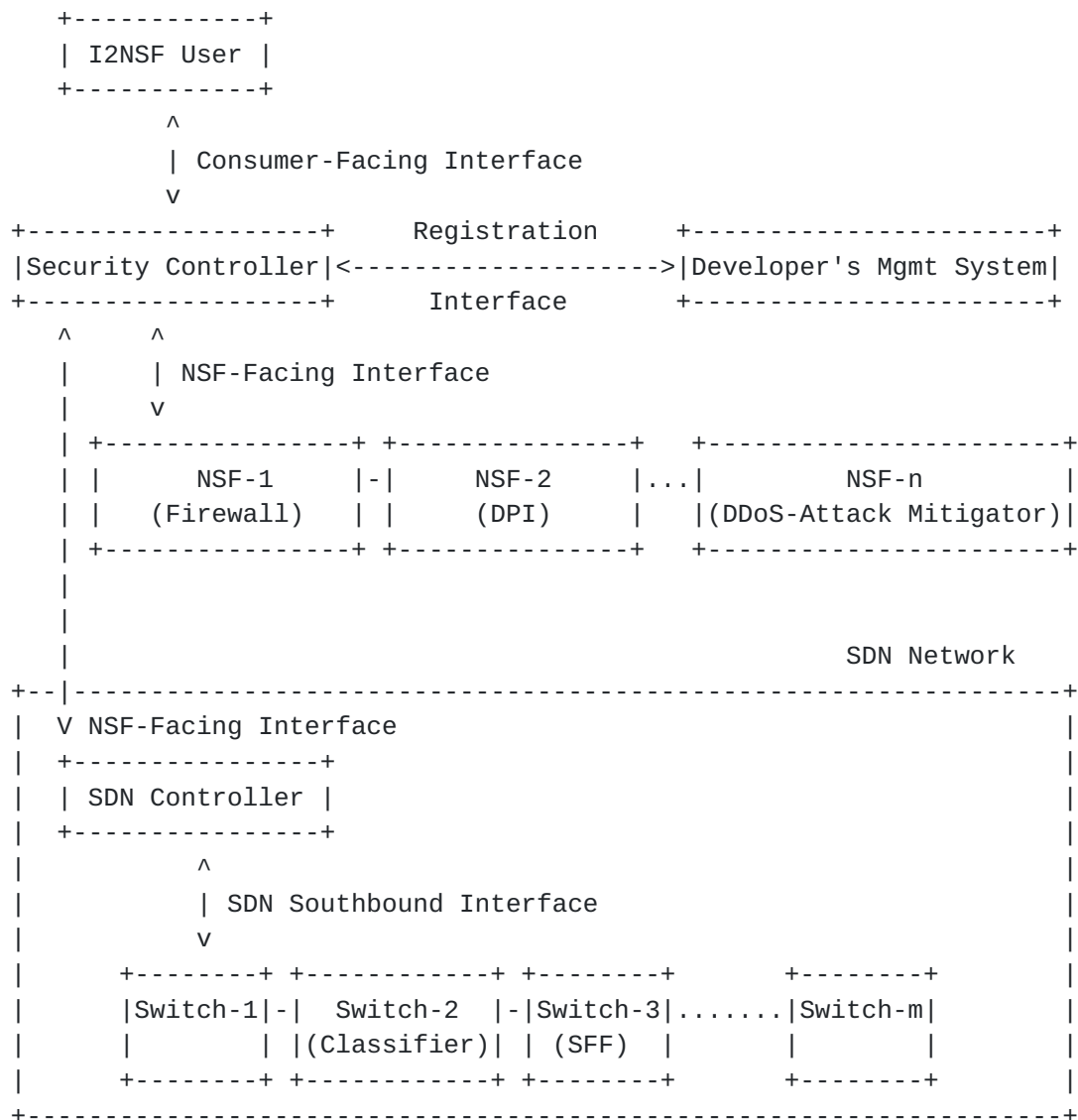
              Figure 3: An I2NSF Framework with SDN Network

   Figure 3 shows an I2NSF framework [RFC8329] with SDN networks to
   support network-based security services. In this system, the
   enforcement of security policy rules is divided into the SDN
   forwarding elements (e.g., switch) and NSFs. Especially, SDN
   forwarding elements enforce simple packet filtering rules that can be
   translated into their packet forwarding rules, whereas NSFs enforce
   NSF-related security rules requiring the security capabilities of the
   NSFs. For this purpose, the Security Controller instructs the SDN
   Controller via NSF-Facing Interface so that SDN forwarding elements
   can perform the required security services with flow tables under the
   supervision of the SDN Controller.

   As an example, let us consider two different types of security rules:

Rule A is a simple packet fltering rule that checks only the IP
address and port number of a given packet, whereas rule B is a time-
consuming packet inspection rule for analyzing whether an attached
file being transmitted over a flow of packets contains malware. Rule
A can be translated into packet forwarding rules of SDN forwarding
elements and thus be enforced by these elements. In contrast, rule B
cannot be enforced by forwarding elements, but it has to be enforced
by NSFs with anti-malware capability. Specifically, a flow of
packets is forwarded to and reassembled by an NSF to reconstruct the
attached file stored in the flow of packets. The NSF then analyzes
the file to check the existence of malware. If the file contains
malware, the NSF drops the packets.

In an I2NSF framework with SDN, the Security Controller can analyze
given security policy rules and automatically determine which of the
given security policy rules should be enforced by SDN forwarding
elements and which should be enforced by NSFs. If some of the given
rules requires security capabilities that can be provided by SDN
forwarding elements, then the Security Controller instructs the SDN
Controller via NSF-Facing Interface so that SDN forwarding elements
can enforce those security policy rules with flow tables under the
supervision of the SDN Controller. Or if some rules require security
capabilities that cannot be provided by SDN forwarding elements but
by NSFs, then the Security Controller instructs relevant NSFs to
enforce those rules.

For the support of SFC in the I2NSF framework with SDN, as shown in
Figure 3, network forwarding elements (e.g., switch) can play the
role of either SFC Classifier or SFF, which are explained in
Section 5. Classifier and SFF have an NSF-Facing Interface with
Security Controller. This interface is used to update security
service function chaining information for traffic flows. For
example, when it needs to update an SFP for a traffic flow in an SDN
network, as shown in Figure 3, SFF (denoted as Switch-3) asks
Security Controller to update the SFP for the traffic flow (needing
another security service as an NSF) via NSF-Facing Interface. This
update lets Security Controller ask Classifier (denoted as Switch-2)
to update the mapping between the traffic flow and SFP in Classifier
via NSF-Facing Interface.

The following subsections introduce three use cases for cloud-based
security services: (i) firewall system, (ii) deep packet inspection
system, and (iii) attack mitigation system. [RFC8192]

6.1. Firewall: Centralized Firewall System

A centralized network firewall can manage each network resource and
apply common rules to individual network elements (e.g., switch).

The centralized network firewall controls each forwarding element, and firewall rules can be added or deleted dynamically.

The procedure of firewall operations in this system is as follows:

1. A switch forwards an unknown flow's packet to one of the SDN Controllers.

2. The SDN Controller forwards the unknown flow's packet to an appropriate security service application, such as the Firewall.

3. The Firewall analyzes, typically, the headers and contents of the packet.

4. If the Firewall regards the packet as a malicious one with a suspicious pattern, it reports the malicious packet to the SDN Controller.

5. The SDN Controller installs new rules (e.g., drop packets with the suspicious pattern) into underlying switches.

6. The suspected packets are dropped by these switches.

Existing SDN protocols can be used through standard interfaces between the firewall application and switches [RFC7149][ITU-T.Y.3300][ONF-OpenFlow] [ONF-SDN-Architecture].

Legacy firewalls have some challenges such as the expensive cost, performance, management of access control, establishment of policy, and packet-based access mechanism. The proposed framework can resolve the challenges through the above centralized firewall system based on SDN as follows:

o Cost: The cost of adding firewalls to network resources such as routers, gateways, and switches is substantial due to the reason that we need to add firewall on each network resource. To solve this, each network resource can be managed centrally such that a single firewall is manipulated by a centralized server.

o Performance: The performance of firewalls is often slower than the link speed of network interfaces. Every network resource for firewall needs to check firewall rules according to network conditions. Firewalls can be adaptively deployed among network switches, depending on network conditions in the framework.

o The management of access control: Since there may be hundreds of network resources in a network, the dynamic management of access control for security services like firewall is a challenge. In

the framework, firewall rules can be dynamically added for new
malware.

o The establishment of policy: Policy should be established for each
   network resource. However, it is difficult to describe what flows
   are permitted or denied for firewall within a specific
   organization network under management. Thus, a centralized view
   is helpful to determine security policies for such a network.

o Packet-based access mechanism: Packet-based access mechanism is
   not enough for firewall in practice since the basic unit of access
   control is usually users or applications. Therefore, application
   level rules can be defined and added to the firewall system
   through the centralized server.

6.2. Deep Packet Inspection: Centralized VoIP/VoLTE Security System

A centralized VoIP/VoLTE security system can monitor each VoIP/VoLTE
flow and manage VoIP/VoLTE security rules, according to the
configuration of a VoIP/VoLTE security service called VoIP Intrusion
Prevention System (IPS). This centralized VoIP/VoLTE security system
controls each switch for the VoIP/VoLTE call flow management by
manipulating the rules that can be added, deleted or modified
dynamically.

The centralized VoIP/VoLTE security system can cooperate with a
network firewall to realize VoIP/VoLTE security service.
Specifically, a network firewall performs basic security checks of an
unknown flow's packet observed by a switch. If the network firewall
detects that the packet is an unknown VoIP call flow's packet that
exhibits some suspicious patterns, then it triggers the VoIP/VoLTE
security system for more specialized security analysis of the
suspicious VoIP call packet.

The procedure of VoIP/VoLTE security operations in this system is as
follows:

1. A switch forwards an unknown flow's packet to the SDN Controller,
    and the SDN Controller further forwards the unknown flow's packet
    to the Firewall for basic security inspection.

2. The Firewall analyzes the header fields of the packet, and
    figures out that this is an unknown VoIP call flow's signal
    packet (e.g., SIP packet) of a suspicious pattern.

3. The Firewall triggers an appropriate security service function,
    such as VoIP IPS, for detailed security analysis of the
    suspicious signal packet. In order for this triggering of VoIP

IPS to be served, the suspicious packet is sent to the Service
Function Forwarder (SFF) that is usually a switch in an SDN
network, as shown in Figure 3. The SFF forwards the suspicious
signal packet to the VoIP IPS.

4. The VoIP IPS analyzes the headers and contents of the signal
    packet, such as calling number and session description headers
    [RFC4566].

5. If, for example, the VoIP IPS regards the packet as a spoofed
    packet by hackers or a scanning packet searching for VoIP/VoLTE
    devices, it drops the packet. In addition, the VoIP IPS requests
    the SDN Controller to block that packet and the subsequent
    packets that have the same call-id.

6. The SDN Controller installs new rules (e.g., drop packets) into
    underlying switches.

7. The illegal packets are dropped by these switches.

Existing SDN protocols can be used through standard interfaces
between the VoIP IPS application and switches [RFC7149][ITU-T.Y.3300]
[ONF-OpenFlow][ONF-SDN-Architecture].

Legacy hardware based VoIP IPS has some challenges, such as
provisioning time, the granularity of security, expensive cost, and
the establishment of policy. The I2NSF framework can resolve the
challenges through the above centralized VoIP/VoLTE security system
based on SDN as follows:

o Provisioning: The provisioning time of setting up a legacy VoIP
   IPS to network is substantial because it takes from some hours to
   some days. By managing the network resources centrally, VoIP IPS
   can provide more agility in provisioning both virtual and physical
   network resources from a central location.

o The granularity of security: The security rules of a legacy VoIP
   IPS are compounded considering the granularity of security. The
   proposed framework can provide more granular security by
   centralizing security control into a switch controller. The VoIP
   IPS can effectively manage security rules throughout the network.

o Cost: The cost of adding VoIP IPS to network resources, such as
   routers, gateways, and switches is substantial due to the reason
   that we need to add VoIP IPS on each network resource. To solve
   this, each network resource can be managed centrally such that a
   single VoIP IPS is manipulated by a centralized server.

   o The establishment of policy: Policy should be established for each
      network resource. However, it is difficult to describe what flows
      are permitted or denied for VoIP IPS within a specific
      organization network under management. Thus, a centralized view
      is helpful to determine security policies for such a network.

6.3. Attack Mitigation: Centralized DDoS-attack Mitigation System

   A centralized DDoS-attack mitigation can manage each network resource
   and manipulate rules to each switch through a common server for DDoS-
   attack mitigation (called DDoS-attack Mitigator). The centralized
   DDoS-attack mitigation system defends servers against DDoS attacks
   outside the private network, that is, from public networks.

   Servers are categorized into stateless servers (e.g., DNS servers)
   and stateful servers (e.g., web servers). For DDoS-attack
   mitigation, traffic flows in switches are dynamically configured by
   traffic flow forwarding path management according to the category of
   servers [AVANT-GUARD]. Such a managenent should consider the load
   balance among the switches for the defense against DDoS attacks.

   The procedure of DDoS-attack mitigation operations in this system is
   as follows:

   1. A Switch periodically reports an inter-arrival pattern of a
       flow's packets to one of the SDN Controllers.

   2. The SDN Controller forwards the flow's inter-arrival pattern to
       an appropriate security service application, such as DDoS-attack
       Mitigator.

   3. The DDoS-attack Mitigator analyzes the reported pattern for the
       flow.

   4. If the DDoS-attack Mitigator regards the pattern as a DDoS
       attack, it computes a packet dropping probability corresponding
       to suspiciousness level and reports this DDoS-attack flow to the
       SDN Controller.

   5. The SDN Controller installs new rules into switches (e.g.,
       forward packets with the suspicious inter-arrival pattern with a
       dropping probability).

   6. The suspicious flow's packets are randomly dropped by switches
       with the dropping probability.

   For the above centralized DDoS-attack mitigation system, the existing
   SDN protocols can be used through standard interfaces between the

DDoS-attack mitigator application and switches [RFC7149]
[ITU-T.Y.3300][ONF-OpenFlow][ONF-SDN-Architecture].

The centralized DDoS-attack mitigation system has challenges similar
to the centralized firewall system. The proposed framework can
resolve the challenges through the above centralized DDoS-attack
mitigation system based on SDN as follows:

o Cost: The cost of adding DDoS-attack mitigators to network
   resources such as routers, gateways, and switches is substantial
   due to the reason that we need to add DDoS-attack mitigator on
   each network resource. To solve this, each network resource can
   be managed centrally such that a single DDoS-attack mitigator is
   manipulated by a centralized server.

o Performance: The performance of DDoS-attack mitigators is often
   slower than the link speed of network interfaces. The checking of
   DDoS attacks may reduce the performance of the network interfaces.
   DDoS-attack mitigators can be adaptively deployed among network
   switches, depending on network conditions in the framework.

o The management of network resources: Since there may be hundreds
   of network resources in an administered network, the dynamic
   management of network resources for performance (e.g., load
   balancing) is a challenge for DDoS-attack mitigation. In the
   framework, as dynamic network resource management, traffic flow
   forwarding path management can handle the load balancing of
   network switches [AVANT-GUARD]. With this management, the current
   and near-future workload can be spread among the network switches
   for DDoS-attack mitigation. In addition, DDoS-attack mitigation
   rules can be dynamically added for new DDoS attacks.

o The establishment of policy: Policy should be established for each
   network resource. However, it is difficult to describe what flows
   are permitted or denied for new DDoS-attacks (e.g., DNS reflection
   attack) within a specific organization network under management.
   Thus, a centralized view is helpful to determine security policies
   for such a network.

So far this section has described the procedure and impact of the
three use cases for network-based security services using the I2NSF
framework with SDN networks. To support these use cases in the
proposed data-driven security service framework, YANG data models
described in [consumer-facing-inf-dm], [nsf-facing-inf-dm], and
[registration-inf-dm] can be used as Consumer-Facing Interface, NSF-
Facing Interface, and Registration Interface, respectively, along
with RESTCONF [RFC8040] and NETCONF [RFC6241].

7. I2NSF Framework with NFV

   This section discusses the implementation of the I2NSF framework
   using Network Functions Virtualization (NFV).

```
                                          +-------------------+
   +-----------------------------------------+ | ---------------   |
   |           I2NSF User (OSS/BSS)          | | | NFV         |   |
   +------+----------------------------------+ | | Orchestrator +-+ |
          |  Consumer-Facing Interface       | -----+---------- | |
   +------|----------------------------------+ |     |          | |
   | -----+---------- (a) ----------------   | |  ----+-----     | |
   | | Security    +-------+ Developer's  |  | |  |        |     | |
   | |Controller(EM)|      |Mgmt System(EM)| +-(b)-+ VNFM(s)|     | |
   | -----+----------      ----------------  | |  |        |     | |
   |       | NSF-Facing Interface            | |  ----+-----     | |
   |   ----+-----    ----+-----    ----+-----| |      |          | |
   |  |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 |
   |                                            | |        (MANO)      |
   +-----------------------------------------+ +-------------------+
```

   (a) = Registration Interface
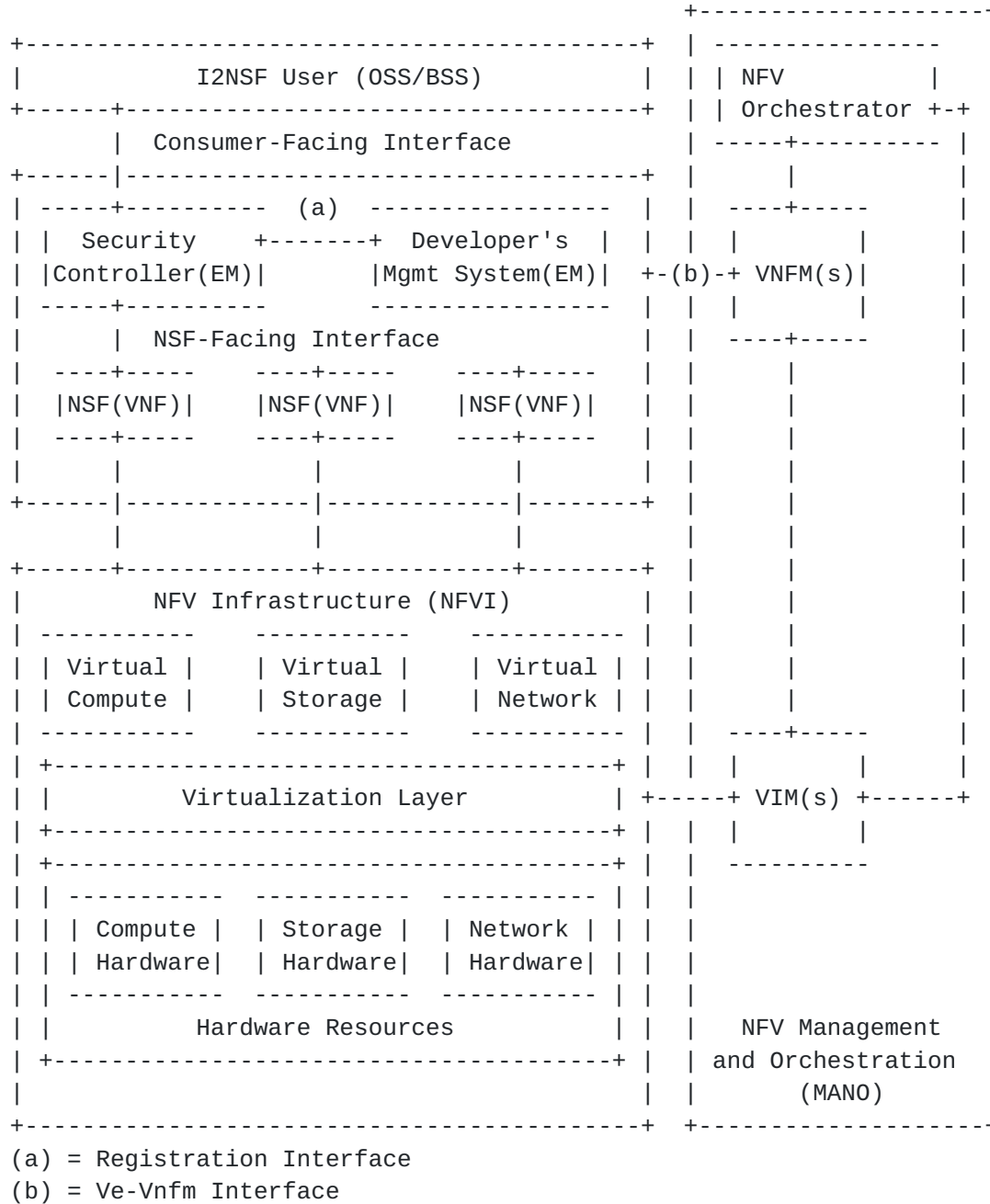   (b) = Ve-Vnfm Interface

      Figure 4: I2NSF Framework Implementation with respect to the NFV
                   Reference Architectural Framework

   NFV is a promising technology for improving the elasticity and
   efficiency of network resource utilization. In NFV environments,

NSFs can be deployed in the forms of software-based virtual instances
rather than physical appliances. Virtualizing NSFs makes it possible
to rapidly and flexibly respond to the amount of service requests by
dynamically increasing or decreasing the number of NSF instances.
Moreover, NFV technology facilitates flexibly including or excluding
NSFs from multiple security solution vendors according to the changes
on security requirements. In order to take advantages of the NFV
technology, the I2NSF framework can be implemented on top of an NFV
infrastructure as show in Figure 4.

Figure 4 shows an I2NSF framework implementation based on the NFV
reference architecture that the European Telecommunications Standards
Institute (ETSI) defines [ETSI-NFV]. The NSFs are deployed as
virtual network functions (VNFs) in Figure 4. The Developer's
Management System (DMS) in the I2NSF framework is responsible for
registering capability information of NSFs into the Security
Controller. Those NSFs are created or removed by a virtual network
functions manager (VNFM) in the NFV architecture that performs the
life-cycle management of VNFs. The Security Controller controls and
monitors the configurations (e.g., function parameters and security
policy rules) of VNFs. Both the DMS and Security Controller can be
implemented as the Element Managements (EMs) in the NFV architecture.
Finally, the I2NSF User can be implemented as OSS/BSS (Operational
Support Systems/Business Support Systems) in the NFV architecture
that provides interfaces for users in the NFV system.

The operation procedure in the I2NSF framework based on the NFV
architecture is as follows:

1. The VNFM has a set of virtual machine (VM) images of NSFs, and
    each VM image can be used to create an NSF instance that provides
    a set of security capabilities. The DMS initially registers a
    mapping table of the ID of each VM image and the set of
    capabilities that can be provided by an NSF instance created from
    the VM image into the Security Controller.

2. If the Security Controller does not have any instantiated NSF
    that has the set of capabilities required to meet the security
    requirements from users, it searches the mapping table
    (registered by the DMS) for the VM image ID corresponding to the
    required set of capabilities.

3. The Security Controller requests the DMS to instantiate an NSF
    with the VM image ID via VNFM.

4. When receiving the instantiation request, the VNFM first asks the
    NFV orchestrator for the permission required to create the NSF
    instance, requests the VIM to allocate resources for the NSF

instance, and finally creates the NSF instance based on the
allocated resources.

5. Once the NSF instance has been created by the VNFM, the DMS
   performs the initial configurations of the NSF instance and then
   notifies the Security Controller of the NSF instance.

6. After being notified of the created NSF instance, the Security
   Controller delivers low-level security policy rules to the NSF
   instance for policy enforcement.

We can conclude that the I2NSF framework can be implemented based on
the NFV architecture framework. Note that the registration of the
capabilities of NSFs is performed through the Registration Interface
and the lifecycle management for NSFs (VNFs) is performed through the
Ve-Vnfm interface between the DMS and VNFM, as shown in Figure 4.
More details about the I2NSF framework based on the NFV reference
architecture are described in [i2nsf-nfv-architecture].

8. Security Considerations

The same security considerations for the I2NSF framework [RFC8329]
are applicable to this document.

This document shares all the security issues of SDN that are
specified in the "Security Considerations" section of [ITU-T.Y.3300].

9. 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 work has been partially supported by the European Commission
under Horizon 2020 grant agreement no. 700199 "Securing against
intruders and other threats through a NFV-enabled environment
(SHIELD)". This support does not imply endorsement.

10. Contributors

I2NSF is a group effort. I2NSF has had a number of contributing
authors. The following are considered co-authors:

o Hyoungshick Kim (Sungkyunkwan University)

o Jinyong Tim Kim (Sungkyunkwan University)

o Hyunsik Yang (Soongsil University)

o Younghan Kim (Soongsil University)

o Jung-Soo Park (ETRI)

o Se-Hui Lee (Korea Telecom)

o Mohamed Boucadair (Orange)

11. Informative References

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

[RFC6020]   Bjorklund, M., "YANG - A Data Modeling
                       Language for the Network Configuration
                       Protocol (NETCONF)", RFC 6020,
                       October 2010.

[RFC6241]   Enns, R., Bjorklund, M., Schoenwaelder, J.,
                       and A. Bierman, "Network Configuration
                       Protocol (NETCONF)", RFC 6241, June 2011.

[RFC8040]   Bierman, A., Bjorklund, M., and K. Watsen,
                       "RESTCONF Protocol", RFC 8040,
                       January 2017.

[consumer-facing-inf-im] Kumar, R., Lohiya, A., Qi, D., Bitar, N.,
                       Palislamovic, S., Xia, L., and J. Jeong,
                       "Information Model for Consumer-Facing
                       Interface to Security Controller", draft-
                       kumar-i2nsf-client-facing-interface-im-07
                       (work in progress), July 2018.

[consumer-facing-inf-dm] Jeong, J., Kim, E., Ahn, T., Kumar, R., and
                       S. Hares, "I2NSF Consumer-Facing Interface
                       YANG Data Model", draft-ietf-i2nsf-
                       consumer-facing-interface-dm-01 (work in
                       progress), July 2018.

[i2nsf-nsf-cap-im]   Xia, L., Strassner, J., Basile, C., and D.
                       Lopez, "Information Model of NSFs
                       Capabilities",

draft-ietf-i2nsf-capability-02 (work in
                             progress), July 2018.

     [policy-translation]   Yang, J., Jeong, J., and J. Kim, "Security
                             Policy Translation in Interface to Network
                             Security Functions", draft-yang-i2nsf-
                             security-policy-translation-01 (work in
                             progress), July 2018.

     [nsf-facing-inf-dm]   Kim, J., Jeong, J., Park, J., Hares, S.,
                             and Q. Lin, "I2NSF Network Security
                             Function-Facing Interface YANG Data Model",
                             draft-ietf-i2nsf-nsf-facing-interface-data-
                             model-01 (work in progress), July 2018.

     [registration-inf-dm]   Hyun, S., Jeong, J., Roh, T., Wi, S., and
                             J. Park, "I2NSF Registration Interface YANG
                             Data Model",
                             draft-hyun-i2nsf-registration-dm-06 (work
                             in progress), July 2018.

     [nsf-triggered-steering] Hyun, S., Jeong, J., Park, J., and S.
                             Hares, "Service Function Chaining-Enabled
                             I2NSF Architecture",
                             draft-hyun-i2nsf-nsf-triggered-steering-06
                             (work in progress), July 2018.

     [i2nsf-nfv-architecture] Yang, H. and Y. Kim, "I2NSF on the NFV
                             Reference Architecture",
                             draft-yang-i2nsf-nfv-architecture-02 (work
                             in progress), June 2018.

     [RFC7149]   Boucadair, M. and C. Jacquenet, "Software-
                             Defined Networking: A Perspective from
                             within a Service Provider Environment",
                             RFC 7149, March 2014.

     [ITU-T.Y.3300]   Recommendation ITU-T Y.3300, "Framework of
                             Software-Defined Networking", June 2014.

     [ONF-OpenFlow]   ONF, "OpenFlow Switch Specification
                             (Version 1.4.0)", October 2013.

     [ONF-SDN-Architecture] ONF, "SDN Architecture", June 2014.

     [ITU-T.X.1252]   Recommendation ITU-T X.1252, "Baseline
                             Identity Management Terms and Definitions",
                             April 2010.

[ITU-T.X.800]   Recommendation ITU-T X.800, "Security
                         Architecture for Open Systems
                         Interconnection for CCITT Applications",
                         March 1991.

[AVANT-GUARD]   Shin, S., Yegneswaran, V., Porras, P., and
                         G. Gu, "AVANT-GUARD: Scalable and Vigilant
                         Switch Flow Management in Software-Defined
                         Networks", ACM CCS, November 2013.

[ETSI-NFV]   ETSI GS NFV 002 V1.1.1, "Network Functions
                         Virtualization (NFV); Architectural
                         Framework", October 2013.

[RFC4566]   Handley, M., Jacobson, V., and C. Perkins,
                         "SDP: Session Description Protocol",
                         RFC 4566, July 2006.

[i2nsf-terminology]   Hares, S., Strassner, J., Lopez, D., Xia,
                         L., and H. Birkholz, "Interface to Network
                         Security Functions (I2NSF) Terminology",
                         draft-ietf-i2nsf-terminology-06 (work in
                         progress), July 2018.

[opsawg-firewalls]   Baker, F. and P. Hoffman, "On Firewalls in
                         Internet Security",
                         draft-ietf-opsawg-firewalls-01 (work in
                         progress), October 2012.

[RFC8192]   Hares, S., Lopez, D., Zarny, M., Jacquenet,
                         C., Kumar, R., and J. Jeong, "Interface to
                         Network Security Functions (I2NSF): Problem
                         Statement and Use Cases", RFC 8192,
                         July 2017.

[RFC7665]   Halpern, J. and C. Pignataro, "Service
                         Function Chaining (SFC) Architecture",
                         RFC 7665, October 2015.

[RFC8300]   Quinn, P., Elzur, U., and C. Pignataro,
                         "Network Service Header (NSH)", RFC 8300,
                         January 2018.

Appendix A. Changes from draft-ietf-i2nsf-applicability-06

The following change has been made from
draft-ietf-i2nsf-applicability-06:

o Add the acknowledgment to the EU H2020 project SHIELD.

Authors' Addresses

   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

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

   Phone: +82 62 230 7473
   EMail: shyun@chosun.ac.kr

   Tae-Jin Ahn
   Korea Telecom
   70 Yuseong-Ro, Yuseong-Gu
   Daejeon 305-811
   Republic of Korea

   Phone: +82 42 870 8409
   EMail: taejin.ahn@kt.com

   Susan Hares
   Huawei
   7453 Hickory Hill
   Saline, MI 48176
   USA

   Phone: +1-734-604-0332
   EMail: shares@ndzh.com

      Diego R. Lopez
      Telefonica I+D
      Jose Manuel Lara, 9
      Seville,   41013
      Spain

      Phone: +34 682 051 091
      EMail: diego.r.lopez@telefonica.com