

Network Working Group
Internet-Draft
Intended status: Informational
Expires: September 6, 2018

J. Jeong
S. Hyun
Sungkyunkwan University
T. Ahn
Korea Telecom
S. Hares
Huawei
D. Lopez
Telefonica I+D
March 5, 2018

Applicability of Interfaces to Network Security Functions to Network-
Based Security Services
draft-ietf-i2nsf-applicability-02

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

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 September 6, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of

Internet-Draft

I2NSF Applicability

March 2018

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Terminology	3
3.	I2NSF Framework	4
3.1.	Time-dependent Web Access Control Service	5
4.	I2NSF Framework with SDN	7
4.1.	Firewall: Centralized Firewall System	10
4.2.	Deep Packet Inspection: Centralized VoIP/VoLTE Security System	11
4.3.	Attack Mitigation: Centralized DDoS-attack Mitigation System	13
5.	Security Considerations	15
6.	Acknowledgments	15
7.	Contributors	15
8.	Informative References	15
Appendix A.	Changes from draft-ietf-i2nsf-applicability-01	19
	Authors' Addresses	19

[1.](#) Introduction

Interface to Network Security Functions (I2NSF) defined 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 NFV environment 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 registers itself to the I2NSF user so that the user can request security services to the Security Controller.

This document describes the applicability of I2NSF framework to network-based security services with a use case of time-dependent web access control. This document also describes integrating I2NSF

framework with Software-Defined Networking (SDN) technology for efficient security services and use cases, such as firewall [[opsawg-firewalls](#)], Deep Packet Inspection (DPI), and Distributed Denial of Service (DDoS) attack mitigation. We implemented the I2NSF

framework based on SDN for these use cases, and the implementation successfully verified the effectiveness of the I2NSF framework.

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-im](#)], [[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. These rules can be managed dynamically by a centralized server for firewall. SDN can work as a network-based firewall system through a standard interface between an SDN switch and a firewall function as a virtual network function (VNF).
- o Centralized VoIP Security System: A centralized security system that handles the security functions required for VoIP and VoLTE services. SDN can work as a network-based security system through

a standard interface between an SDN switch and a VoIP/VoLTE security function as a VNF.

- 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. These rules can be managed dynamically by a centralized server for DDoS-attack mitigation. The SDN controller and switches can cooperatively work as a network-based firewall system through a standard interface between an SDN switch and a firewall function as a VNF running in the SDN controller.

3. I2NSF Framework

This section describes an I2NSF framework and its use case. Figure 1 shows an I2NSF framework [[RFC8329](#)] to support network-based security services. As shown in Figure 1, I2NSF User can use security functions by delivering high-level security policies, which specify security requirements 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. 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 NSFs are enabled as Virtual Network Functions (VNFs) on top of virtual machines through Network Functions Virtualization (NFV) [[ETSI-NFV](#)]. In addition, the Security Controller uses the I2NSF Registration Interface [[registration-inf-im](#)] [[registration-inf-dm](#)] to communicate with Developer's Management System (called Developer's Mgmt System) for registering (or deregistering) the developer's NSFs into (or from) the NFV system using the I2NSF framework.

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.

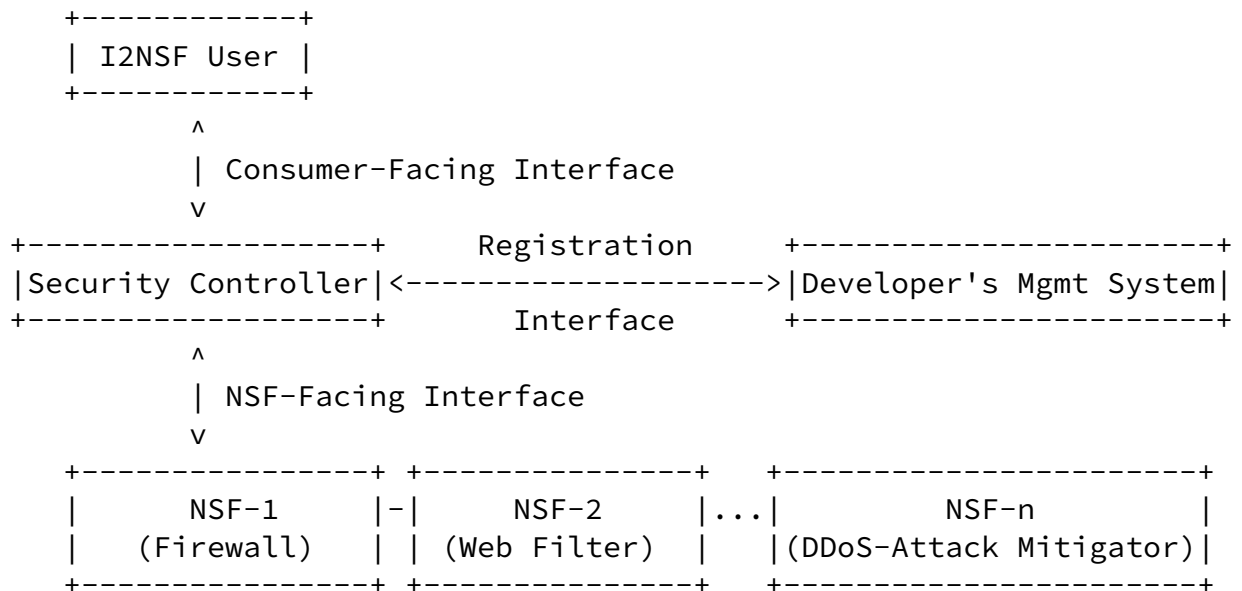


Figure 1: I2NSF Framework

The NSF-Facing Interface between 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 Mgmt 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 service function chaining (SFC) techniques for the I2NSF architecture described in [[nsf-triggered-steering](#)].

The following describes a security service scenario using the I2NSF framework.

[3.1](#). Time-dependent Web Access Control Service

This service scenario assumes that an enterprise network administrator wants to control the staff members' access to Facebook during business hours. The following is an example high-level security policy rule that the administrator requests: Block the staff members' access to Facebook 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 facebook.com 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 Registration interface. Based on this information, the Security Controller identifies NSFs that can perform the IP address and port number inspection and URL inspection. 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 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 facebook.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 Fackbook.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. Service Function Chaining (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 Facebook.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 Facebook during business hours is

blocked.

4. 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 the network switches 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.

Figure 2 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 switches and NSFs. Especially, SDN switches 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 Switch Controller via NSF-Facing Interface so that SDN switches can perform the required security services with flow tables under the supervision of the Switch Controller (i.e., SDN Controller).

As an example, let us consider two different types of security rules: Rule A is a simple packet filtering 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 switches and thus be enforced by the switches. In contrast, rule B cannot be enforced by switches, but it can 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 switches and

which should be enforced by NSFs. If some of the given rules requires security capabilities that can be provided by SDN switches, then the Security Controller instructs the Switch Controller via NSF-Facing Interface so that SDN switches can enforce those security policy rules with flow tables under the supervision of the Switch Controller (i.e., SDN Controller). Or if some rules require security capabilities that can be provided by not SDN switches but NSFs, then the Security Controller instructs relevant NSFs to enforce those rules.

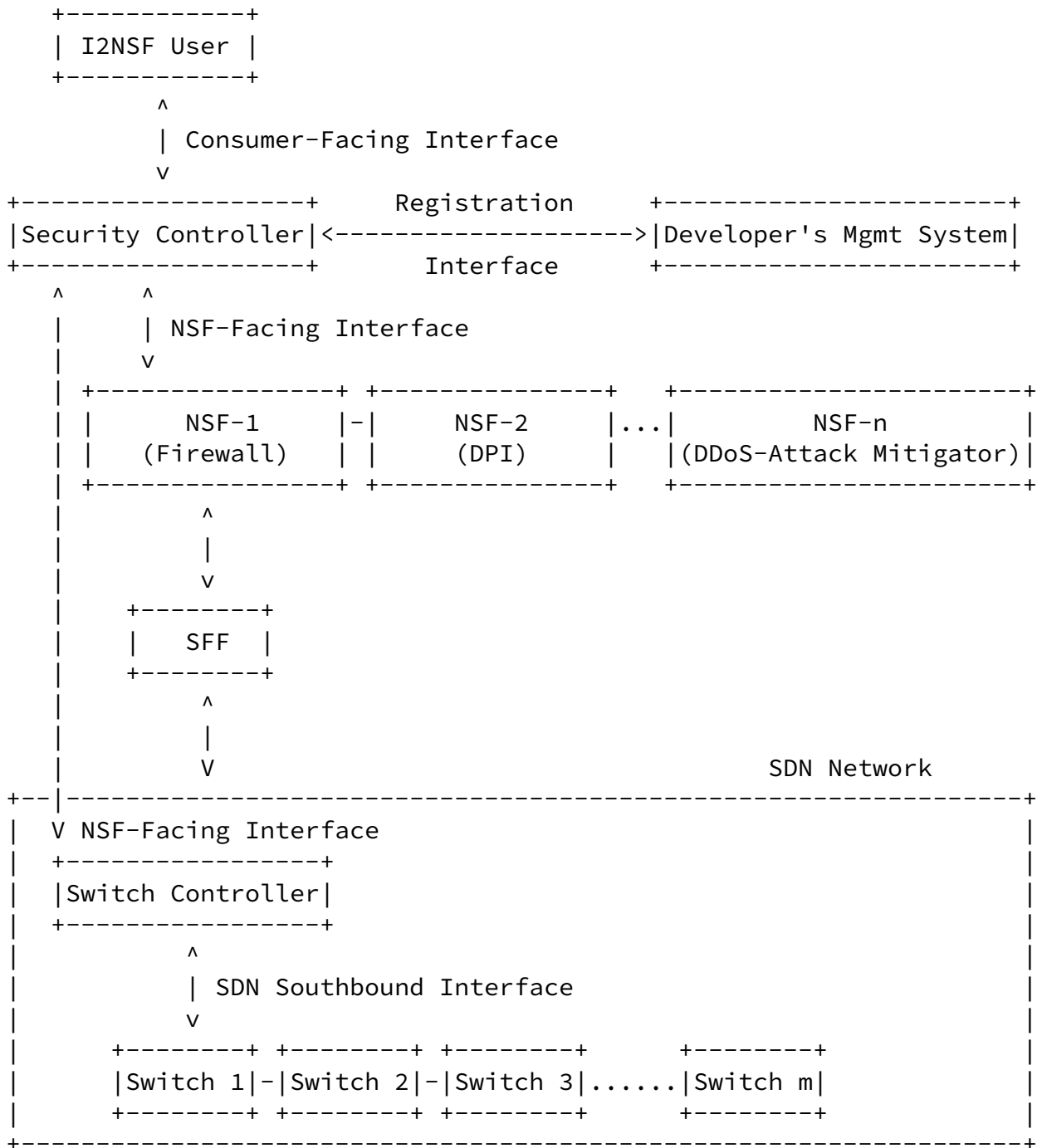


Figure 2: An I2NSF Framework with SDN Network

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](#)]

[4.1](#). Firewall: Centralized Firewall System

A centralized network firewall can manage each network resource and firewall rules can be managed flexibly by a centralized server for firewall (called Firewall). The centralized network firewall controls each switch for the network resource management and the 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 Switch Controllers.
2. The Switch 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 Switch Controller.
5. The Switch 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.

[4.2.](#) 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 controlled by a centralized server for VoIP/VoLTE security service called VoIP Intrusion Prevention System (IPS). The 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.

A 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 Switch Controller, and the Switch 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. That is, the firewall sends the packet to the Service Function Forwarder (SFF) in the I2NSF framework [[nsf-triggered-steering](#)], as shown in Figure 2. 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 Switch Controller to block that packet and the subsequent packets that have the same call-id.
6. The Switch 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.

4.3. 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 centralized server for DDoS-attack mitigation (called DDoS-attack Mitigator). The centralized DDoS-attack mitigation system defends servers against

DDoS attacks outside private network, that is, from public network.

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 management 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 Switch Controllers.
2. The Switch 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 Switch Controller.
5. The Switch 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 document 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](#)].

5. Security Considerations

The I2NSF framework with SDN networks in this document is derived from the I2NSF framework [[RFC8329](#)], so the security considerations of the I2NSF framework should be included in this document. Therefore, proper secure communication channels should be used the delivery of control or management messages among the components in the proposed framework.

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

6. 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).

7. 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 Jung-Soo Park (ETRI)
- o Se-Hui Lee (Korea Telecom)
- o Mohamed Boucadair (Orange)

8. Informative References

[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.

[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-00](#) (work in progress), March 2018.

[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-04](#) (work in progress), October 2017.

[ETSI-NFV]

ETSI GS NFV 002 V1.1.1, "Network Functions Virtualisation (NFV); Architectural Framework", October 2013.

[i2nsf-nsf-cap-im]

Xia, L., Strassner, J., Basile, C., and D. Lopez, "Information Model of NSFs Capabilities", [draft-ietf-i2nsf-capability-00](#) (work in progress), September 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-05](#) (work in progress), January 2018.

[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.

[ITU-T.Y.3300]

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

[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-00](#) (work in progress), March 2018.

Internet-Draft

I2NSF Applicability

March 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-05](#) (work in progress), March 2018.

[ONF-OpenFlow]

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

[ONF-SDN-Architecture]

ONF, "SDN Architecture", June 2014.

[opsawg-firewalls]

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

[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-03](#) (work in progress), March 2018.

[registration-inf-im]

Hyun, S., Jeong, J., Roh, T., Wi, S., and J. Park, "I2NSF Registration Interface Information Model", [draft-hyun-i2nsf-registration-interface-im-04](#) (work in progress), March 2018.

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

[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.

- [RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", [RFC 7149](#), March 2014.
- [RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", [RFC 8040](#), January 2017.

Jeong, et al.

Expires September 6, 2018

[Page 17]

Internet-Draft

I2NSF Applicability

March 2018

- [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.
- [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-ietf-i2nsf-applicability-01](#)

The following changes have been made from [draft-ietf-i2nsf-applicability-01](#):

- o In [Section 4](#), it is clarified what types of security policy rules can be enforced by SDN switches or NSFs in the environment of I2NSF framework with SDN.
- o In [Section 4](#), it is explained what should be done by the Security Controller in order to divide the enforcement of security policy rules into the SDN switches and NSFs in the I2NSF framework with SDN.

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

Phone: +82 31 290 7222
Fax: +82 31 299 6673
EMail: swhyun77@skku.edu
URI: <http://imtl.skku.ac.kr/>

Jeong, et al.

Expires September 6, 2018

[Page 19]

Internet-Draft

I2NSF Applicability

March 2018

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