Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

S. Hyun
S. Woo
Y. Yeo
J. Jeong
Sungkyunkwan University
J. Park
ETRI

October 31, 2016

Registration Interface Information Model
draft-hyun-i2nsf-registration-interface-im-00

Abstract

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

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 4, 2017.

Copyright Notice

Table of Contents

1. Introduction

A number of virtual network security function instances typically exist in Interface to Network Security Functions (I2NSF) framework [i2nsf-framework]. In this environment, it is important to dynamically manage a Network Security Function (NSF) instance pool for efficient resource utilization. For instance, if a certain NSF instance is receiving an excessive amount of traffic beyond its capacity, an additional instance for the same security function should be created. If an NSF instance is idle for a period of time, it would be better to destroy it to avoid resource waste. In addition, the existing information model for NSF facing interface requires an NSF to trigger another type of NSF for further inspection [capability-interface-im]. In this case, if there is no available instance for the latter NSF, a new NSF should be instantiated. Similarly, in order to enforce a security policy from the client, all the required NSF instances should be created.

This document describes the procedures which should be performed on the registration interface between security controller and developer's management system to dynamically manage a pool of NSF instances. It further describes the detailed information which should be exchanged between security controller and developer's management system.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

3. Terminology

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

- Network Security Function (NSF): A function that is responsible for specific treatment of received packets. A Network Security Function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). Sample Network Security Service Functions are as follows: Firewall, Intrusion Prevention/ Detection System (IPS/IDS), Deep Packet Inspection (DPI), Application Visibility and Control (AVC), network virus and malware scanning, sandbox, Data Loss Prevention (DLP), Distributed Denial of Service (DDoS) mitigation and TLS proxy [nsf-triggered-steering].

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

- Network Security Function Profile (NSF Profile): NSF Profile specifies the inspection capabilities of the associated NSF instance. Each NSF instance has its own NSF Profile to specify the type of security service and its resource capacity [nsf-triggered-steering].

4. Objectives

- Efficient network resource utilization through dynamic instantiation of NSFs and load balancing: In I2NSF framework, it is sometimes possible that a specific NSF experiences heavy traffic loads. For example, under DDoS attacks, a huge volume of traffic would be driven to DoS attack mitigator function to cope with the attacks. In this case, we should allocate a large portion of resources to that DoS attack mitigator function by creating a sufficient number of DoS mitigator instances. After the attack is terminated, we should eliminate some of the instances no longer used. In this way, we can achieve efficient resource utilization. For this purpose, it is indispensable to define an information model of registration interface for dynamic instantiation/elimination of NSF instances.

- Creating an NSF instance to serve another NSF's inspection

request: In I2NSF framework, an NSF triggers another type of NSF(s) when the traffic requires further security inspection. The following NSF is determined by previous NSF's inspection result and client's policy. However, if there is no available instance of the latter NSF required by the former NSF, we should be able to create an NSF instance via Developer's Management System (DMS) and registration interface.

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

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

Controller. DMS then notifies Security Controller of those NSF instances via registration interface.

5. Information Model

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

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

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

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

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

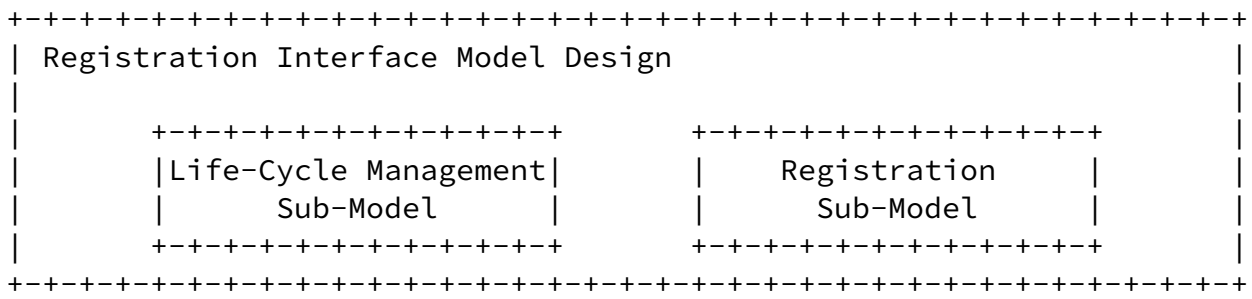

Figure 1: The Registration Interface Model Design

As illustrated in Figure 1, the information model for Registration Interface consists of two sub-models: life-cycle management, registration sub-models. The life-cycle management functionality and the registration functionality use NSF Profile to achieve their goals. In this context, NSF Profile is the capability objects that

describe and/or prescribe inspection capability an NSF instance can provide.

5.1. Life-Cycle Managment Mechanism

For the life-cycle management of NSFs, Security Controller in I2NSF framework requires two types of requests: Instance Creation and Elimination Request Messages. Security Controller sends the request messages to DMS when required. Once receiving the request, DMS conducts creating/eliminating the corresponding NSF instance and responds Security Controller with the results. There are several cases requiring creation of a new NSF instance which provides specific security inspection functionalities and elimination of an existing NSF which is unused for a period of time. For example,

1) When an NSF triggers an advanced inspection of the suspicious traffic via another type of NSF whose instance is currently unavailable in the system.

2) When an NSF instance undergoes an excessive amount of traffic

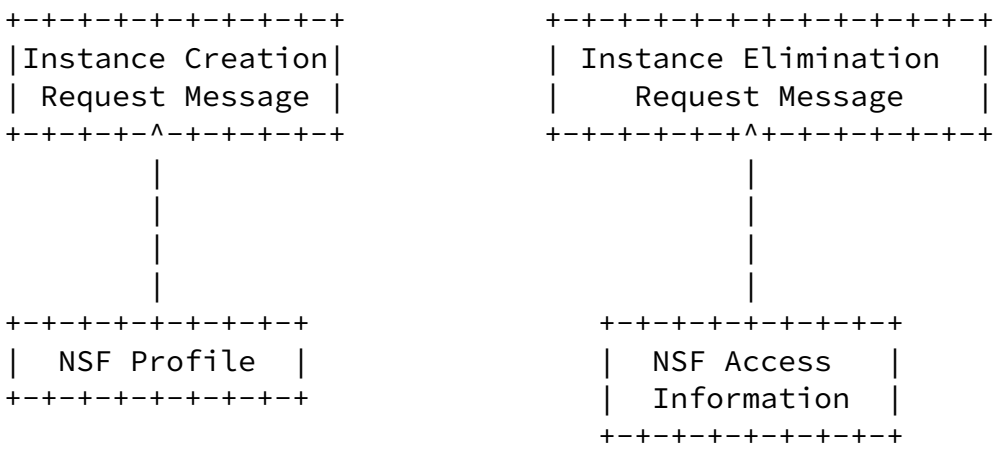

Figure 2: Life-Cycle Management Sub-Model Overview

5.2. Registration Mechanism

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

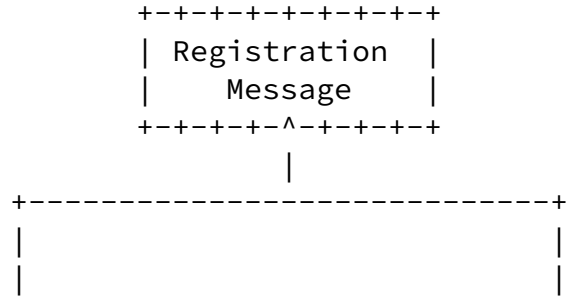

Figure 3: Registration Mechanism Sub-Model Overview

5.3. NSF Access Information

NSF Access Information can contain IPv4 Address, IPv6 Address, Transport Protocol, Port Number etc. In this document, NSF Access Information is used to identify a specific NSF instance (i.e. NSF Access Information is the signature of an NSF instance in the overall system).

5.4. NSF Profile (Capabilities of an NSF instance)

NSF Profile basically refers the inspection capabilities of an NSF instance. As illustrated in Figure 4, it can be split into five capabilities (Content-Matching, Context-Matching, Attack-Mitigation, Action, Performance Capabilities). We share security capabilities which are defined in Section 3 (Overall Analysis of Security Capability) in [capability-interface-im] for the first five capabilities and append one additional capability.

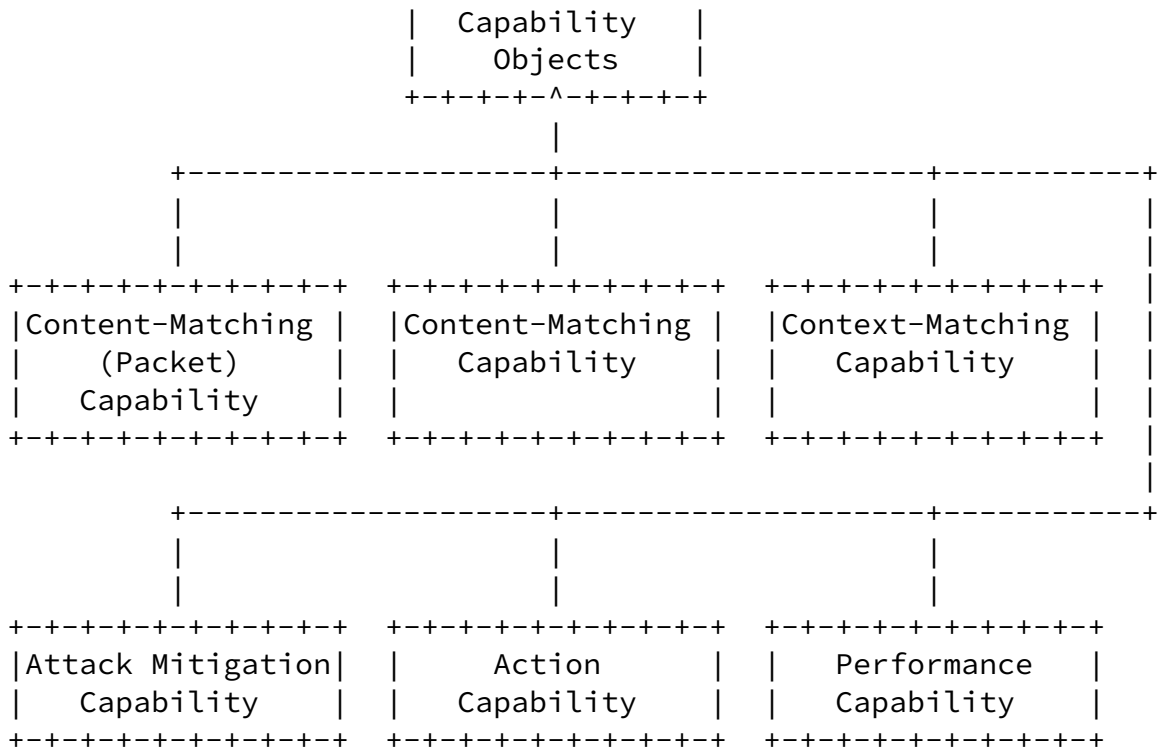

Figure 4: NSF Profile Overview

5.4.1. Packet Content-Matching Capability

Refer to the kind of information or attributes acquired directly from the packet headers or payloads that can be used in the security policy. It can be any fields or attributes in the packet L2/L3/L4 header, or special segment of bytes in the packet payload. [capability-interface-im]

5.4.2. Content-Matching Capability

Content security is another category of security capabilities applied to application layer. Through detecting the contents carried over the traffic in application layer, these capabilities can realize various security functions, such as defending against intrusion, inspecting virus, filtering malicious URL or junk email, blocking illegal web access or malicious data retrieval. [capability-interface-im]

5.4.3. Context-Matching Capability

This capability refers to the content information for the received packets. It can be User, Schedule, Region, Target, State and Direction information. [capability-interface-im]

5.4.4. Attack-Mitigation Capability

This category of security capabilities is used to detect and mitigate various types of network attacks. Today's common network attacks can be classified into the following sets, and each set further consists of a number of specific attacks: [capability-interface-im]

- DDoS attacks:
  - Network layer DDoS attacks: Examples include SYN flood, UDP flood, ICMP flood, IP fragment flood, IPv6 Routing header attack, and IPv6 duplicate address detection attack;
  - Application layer DDoS attacks: Examples include http flood, https flood, cache-bypass http floods, WordPress XML RPC floods, ssl DDoS.
- Single-packet attack:
  - Scanning and sniffing attacks: IP sweep, port scanning, etc
  - Malformed packet attacks: Ping of Death, Teardrop, etc
  - Special packet attacks: Oversized ICMP, Tracert, IP timestamp option packets, etc

5.4.5. Action Capability

NSFs provide security functions by executing various Actions, which at least includes: [capability-interface-im]

- Ingress actions, such as pass, drop, mirroring, etc;
- Egress actions, such as invoke signaling, tunnel encapsulation, packet forwarding and/or transformation;
- Applying a specific Functional Profile or signature (NSF Profile) - The functional profile or signature file defines the security capabilities for content security control and/or attack mitigation control. One goal of I2NSF is to standardize the form and functional interface of those security capabilities while supporting vendor-specific implementations of each.

5.4.6. Performance Capability

This capability represents hardware capacity information such as the

amount of traffic it can process per second. In addition, this information can specify an available amount of each type of resources

such as processing power and memory etc. an NSF instance has.

6. Security Considerations

The information model of the registration interface is based on the I2NSF framework without any architectural changes. Thus, this document shares the security considerations of the I2NSF framwork architecture that are specified in [i2nsf-framework] for the purpose of achieving secure communication among components in the proposed architecture.

7. Acknowledgements

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).

8. References

8.1. Normative References

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

8.2. Informative References

[capability-interface-im] Xia, L., Zhang, D., Lopez, E., Bouthors, N., and L. Fang, "Information Model of Interface to Network Security Functions Capability Interface", draft-xia-i2nsf-capability-interface-im-06 (work in progress), July 2016.

[i2nsf-framework] Lopez, E., Lopez, D., Dunbar, L., Strassner, J., Zhuang, X., Parrott, J., Krishnan, R., Durbha, S., Kumar, R., and A. Lohiya, "Framework for Interface to

Network Security Functions", draft-ietf-i2nsf-framework-03 (work in progress), October 2016.

[i2nsf-terminology] Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-01 (work in progress), October 2016.

[nsf-triggered-steering] Hyun, S., Woo, S., Yeo, Y., Jeong, J., and J. Park, "NSF-Triggered Traffic Steering", draft-hyun-i2nsf-nsf-triggered-steering-in-i2nsf-00 (work in progress).

Authors' Addresses

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/

SangUk Woo
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: suwoo@imtl.skku.ac.kr,
URI: http://imtl.skku.ac.kr/index.php?mid=member_student

YunSuk Yeo
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: yunsuk@imtl.skku.ac.kr,
URI: http://imtl.skku.ac.kr/index.php?mid=member_student

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

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon 305-700
Republic of Korea

Phone: +82 42 860 6514
EMail: pjs@etri.re.kr