Abstract

This document describes a mechanism to control Service Function access to the Network Service Header (NSH). It addresses the Service Function trust issue and provide a method to enforce predefined access control lists to limit Service Function access to Service Function Chain information in the NSH in NSH-based Service Chaining.

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 http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on May 3, 2017.

Copyright Notice

Copyright (c) 2016 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 (http://trustee.ietf.org/license-info) in effect on the date of 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
   1.1. Requirements Language .................................. 2
   1.2. Problem Statement ...................................... 2
   1.3. Definition Of Terms ...................................... 3
2. SF Access Control List ................................. 4
3. Access Control Enforcing Mechanisms ................. 4
4. Consideration for NSH Concealment .................... 7
5. Acknowledgements ........................................... 7
6. IANA Considerations ........................................ 7
7. Security Considerations .................................... 7
8. Normative References ....................................... 7
Authors’ Addresses .............................................. 8

1. Introduction

1.1. 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].

1.2. Problem Statement

SFC Architecture document [RFC7665] defines architectural concepts and core components, including Service Functions (SFs), Service Function Forwarder (SFF), Classifier (CF), SFC Proxy. These terminologies will be used in this document.

It is argued that whether or not we should trust the SFs in SFC. In SFC general use cases, SFs vary from virtual services hosted in general-purpose servers to legacy service functions with dedicated hardware. Most of the time, these SFs are deployed and operated by their service provider, therefore they are highly trusted. Despite of being in isolated and relatively safe service provider networks, SFs are not invulnerable to all security threats. Indeed, there are several reasons that cause the misbehavior of SFs. For instance, they can still be manipulated by multiple types of malware. Furthermore, malfunctioned and misconfigured SFs can have anomaly behavior as well.

Aside from their own SFs, service providers may use SFs from other sources such as third party service providers (in case they want to outsource their SFs), SFs on customer premise, and legacy black-box SFs. In addition, enterprises are also trying to outsource their SFs to service providers, while still manage the SFC by themselves.
Although service providers always have some SLAs for each SF, these SFs need to be verified and security checked frequently.

Even if the SFs are trusted and secured, we still need to concern about the security of transportation layer. Traffic between SFs and forwarding components (SFF, Classifier, SFC proxy) can be harmed by other threats such as man-in-the-middle attacks and spoofing, especially in geographically distributed data centers.

As described in [I-D.ietf-sfc-nsh], NSH can be used to encapsulate SFC information to the packets in NSH-based Service Chaining. There have been considerations about the security of this encapsulation. Problem Statement for Service Function Chaining [RFC7498] and SFC Architecture document [RFC7665] express concerns about SFC Encapsulation security, which emphasize the importance of securing sensitive metadata carried by the encapsulation and state the requirement of an "appropriate protective treatment of NSH information". Specifically, the NSH document [I-D.ietf-sfc-nsh] suggests some options (e.g. IP Sec) to provide NSH metadata authenticity and confidentiality, most of them involve NSH encryption.

Certainly, NSH encryption can provide rather strong security for the SFC metadata in an SFC-enabled domain, but it is also costly. Both header encryption and key distribution require lots of resources and probably cause performance penalties to the SFC. In this document, we describe an inexpensive mechanism to protect the sensitive metadata in NSH from either corrupted SFs or underlay networks security threats in NSH-based Service Chaining.

1.3. Definition Of Terms

- **Access-Controlled Segment (ACS)**: an ACS is an area/field within NSH that carries a piece of sensitive SFC information needed to be protected. The access to this information from SFs should be limited and being controlled.

- **SF Access Control List (SF ACL)**: a list describes the access permission of an SF to each ACS in the packet passing though it. Each SF should have an SF ACL.

- **Ingress Subsequent Classifier (Ingress S-CF)**: a logical classifier located BEFORE an access-controlled SF. S-CFs are responsible for classify packets going into the SF and update their NSH.

- **Egress Subsequent Classifier (Egress S-CF)**: a logical classifier located AFTER an access-controlled SF. S-CFs are responsible for classify packets going out of the SF and update their NSH.
o NSH-state: a set of value/information stored in the NSH of a
  packet at a particular moment. For example, the value of Service
  Index, Service Path Index, Mandatory Context Header 1-4. An NSH-
  state usually consists of some ACS values.

2. SF Access Control List

Currently, without packet encryption, all SFs have full access to the
packets they process and, in particular, the SFC information.
Consequently, an SF can read and modify any unencrypted information
within the NSH during its packet handling process. Depend on what
metadata stored in NSH, a corrupted SF can manipulate thin
information for harmful purposes such as changing service path, SF
spoofing, gathering tenant information.

An SF might not, and in most situations, need not to know all SFC
information to process its incoming packets. An Access Control List
of an SF contains access permissions of the SF to each ACS in the
NSH, including: Service Path Index (SPI), Service Index (SI), and
Metadata (MD). For MD type 1, each of the 4 Mandatory Context Header
(MCH) can be an ACS. MD type 2 has variable length MCH, therefore SF
ACL should be defined according to the MD structure. We propose
three levels of permission to access an ACS:

  o Hidden: the SF cannot view the information in this ACS
  o Read-only: the SF can view, but cannot modify the information in
    this ACS
  o Modify: the SF cannot view and modify the information in this ACS

3. Access Control Enforcing Mechanisms

In this section, we propose a mechanism to enforce SF ACLs in SFC.
The mechanism has two principles:

1. Only give SFs what information they need to access.

2. SFFs cannot control SFs not to modify SFC information, but they
can choose not to accept the modification.

Figure 1 illustrates which components are involved in the mechanism.
Occasionally, an SF is attached to an SFF. However, according [I-
D.ietf-sfc-nsh], SFFs CANNOT perform NSH updating, which is the
essential requirement of this mechanism. Therefore, Ingress/Egress
S-CFs are added and coordinate with SFF to classify packets and
update NSH.
When a packet packet come to an SFF on its way to the next SF in the SFP, the SFF will forward it to the ingress S-CF corresponds to the SF. Next, the ingress S-CF will check the SF ACL to get the SF permission to each ACS. If the SF has Hidden permission to an ACS, the data contained in that ACS will be stored by the S-CF and erased from the packet (i.e. set all byte to zero). As a result, sensitive information will be obscured from the SF, hence reduce the possibility of leaking information. In addition, the ingress S-CF does not store only the erased ACS but also the ACSes that the SF has read-only permission for later use.

After processing the packet, the SF forwards it back to an egress S-CF, which could be the same one as the aforementioned ingress S-CF. This S-CF checks the SF ACL and 1) With Hidden ACS, it gets the stored NSH-state (consists of ACS values) from the ingress S-CF and put it back to the packet, 2) With read-only ACS, it also gets the value from the NSH-state stored in the ingress S-CF and compares to the current value. If the value was changed, the SF had tried to modify the ACS and egress S-CF would recover the ACS from the NSH-state stored by ingress S-CF. Using this method, we can preserve the original SFC information, as well as detect abnormal SF behaviors.

![Figure 1: Controlling SFs access to NSH in SFC architecture](image)
In this mechanism, the ingress and egress S-CF must exchange stored ACS data in either way:

- Shared storage
- Send the data to the SFC control plane

Another key point of this mechanism is how to track the packet between ingress and egress S-CFs in order to recover the appropriate NSH metadata. In detail, a packet encapsulation (including NSH) and payload might be changed completely after it traverses the SF between ingress and egress S-CFs. The egress S-CF must know which NSH-state saved by the ingress SFF corresponds with a packet it receives. Current solutions for this include:

- Reclassification: The packet will be classified (i.e. Using 5-tuple) after it exits the SF and goes to the egress S-CF. The appropriate NSH will be determined based on the classification result. Nevertheless, this approach cannot work with SFs which change 5-tuple (such as NAT)

- Using Metadata: In this approach, the ingress S-CF put a unique ID named NSH-state ID into NSH metadata of the packets. The egress S-CF get the ID from a packet and use it to determine the packet’s original NSH-state. Figure 2 presents an example of NSH having Metadata type 1 with NSH-state ID. In addition, NSH-state ID can be defined per-flow or per-packet. Nevertheless, the disadvantage is that we have to reserve a part of Metadata, which also can be access by SFs, for NSH-state ID.

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Ver | O | C | R | R | R | R | R | R | Length | MD-type=0x1 | Next Protocol |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Service Path Identifier | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| NSH-State ID |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Mandatory Context Header |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Mandatory Context Header |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Mandatory Context Header |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 2: NSH-state ID in NSH Metadata type 1
This mechanism guarantees the consistency of sensitive SFC data within NSH when packets travel from an ingress SFF to an egress SFF through an SF, which means it can eliminate potential threats from the SF as well as the transportation networks between SFs and SFFs.

4. Consideration for NSH Concealment

In previous section, we describe mechanism to obscure ACS in NSH from SFs. However, it is not limited to controlling the access to NSH of only SFs but other components as well. Indeed, the mechanism can be extended to conceal ACS between two any points on the service function path of a packet. In other words, an ingress and an egress SFFs can be any SFF on the SFP, not just the SFF which is directly attached to an SF.

Moreover, the mechanism can be used to perform simple and low-cost NSH encryptions. For example, the ingress SFF will save an ACS value and replace it with another value. The mapping table which map those two values will be given to the egress SFF and the all authorized SFs.

5. Acknowledgements

TBD

6. IANA Considerations

This document does not require any IANA actions.

7. Security Considerations

Secure communications between SFC control plane and components is required, as described in [I-D.ietf-sfc-control-plane], in order to secure access control policies during policy propagation from the control plane to enforcing components (such as SFFs and classifiers).

Furthermore, if the NSH state table of an ingress S-CF is leaked, the controlling mechanism can be easily bypassed by spoofing. Therefore, NSH state exchange process, which is either between CF-control plane or CF-CF, should be secured as well.

8. Normative References

[I-D.ietf-sfc-control-plane]


Authors’ Addresses

Vu Anh Vu
Soongsil University
369 Sangdo-ro
Dongjak-gu, Seoul 06978
South Korea
Email: vuva@dcn.ssu.ac.kr

Younghan Kim
Soongsil University
369 Sangdo-ro
Dongjak-gu, Seoul 06978
South Korea
Email: younghak@ssu.ac.kr