

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

R. Browne
A. Chilikin
Intel
T. Mizrahi
Marvell

October 27, 2016

Network Service Header KPI Stamping
draft-browne-sfc-nsh-kpi-stamp-00.txt

Abstract

This draft describes a method of inserting Key Performance Indicators (KPIs) into Network Service Header (NSH) encapsulated packets or frames on service chains. This method may be used to monitor latency and QoS configuration to identify problems with virtual links (vlinks), Virtual Network Functions (VNFs) or Physical Network Functions (PNFs) on the Rendered Service Path (RSP).

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/lid-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 April 27, 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.....	3
2. Terminology.....	3
2.1. Requirement Language.....	3
2.2. Definition of Terms.....	4
2.3. Abbreviations.....	5
3. NSH KPI Stamping.....	6
3.1. Prerequisites.....	8
3.2. Operation.....	10
3.2.1. Flow Selection.....	11
3.2.2. SCP Interface.....	11
3.3. Performance Considerations.....	12
4. NSH KPI Stamping Encapsulation.....	13
4.1. KPI Stamping Encapsulation (Detection Mode).....	13
4.2. NSH Timestamping Encapsulation (Extended Mode).....	16
4.3. NSH QoS Stamping Encapsulation (Extended Mode).....	19
5. Hybrid Models.....	22
5.1. Targeted VNF Stamp.....	23
6. Fragmentation Considerations.....	23
7. Security Considerations.....	24
8. Open Items for WG Discussion.....	24
9. IANA Considerations.....	25
10. Contributors.....	26
11. Acknowledgments.....	26
12. References.....	26
12.1. Normative References.....	26
12.2. Informative References.....	27

1. Introduction

Network Service Header (NSH), as defined by [NSH], defines a method to insert a service-aware header in between payload and transport headers. This allows a great deal of flexibility and programmability in the forwarding plane allowing user flows to be programmed on-the-fly for the appropriate Service Functions (SFs).

Whilst NSH promises a compelling vista of operational agility for Service Providers, many service providers are concerned about losing service and configuration visibility in the transition from physical appliance SFs to virtualized SFs running in the Network Function Virtualization (NFV) domain. This concern increases when we consider that many service providers wish to run their networks seamlessly in 'hybrid' mode, whereby they wish to mix physical and virtual SFs and run services seamlessly between the two domains.

This draft describes a generic method to monitor and debug service chains in terms of application latency and QoS configuration of the flows within a service chain. This method is compliant with hybrid architectures in which VNFs and PNFs are freely mixed in the service chain. This method also is flexible to monitor the performance and configuration of an entire chain or part thereof as desired. Please refer to [NSH] as background architecture for the method described in this document.

In particular, this draft proposes mechanisms to detect and debug performance issues based on timestamping flows within a chain and to detect and debug QoS configuration on the chain. The method described here is easily extensible to monitoring other KPIs also.

The method described in this draft is not an OAM protocol like [Y.1731] or [Y.1564] for example. As such it does not define new OAM packet types or operation. Rather it monitors the service chain performance and configuration for subscriber payloads and indicates subscriber QoE rather than out-of-band infrastructure metrics.

2. Terminology

2.1. Requirement 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].

2.2. Definition of Terms

Classification: Locally instantiated policy and customer/network/service profile matching of traffic flows for identification of appropriate outbound forwarding actions.

First Stamping Node (FSN): Mark packets correctly. Must understand 5 tuple information in order to match Stamping Controller flow table.

Last Stamping Node (LSN): Reads all MD & export to system performance statistics agent or repository. Should also send NSH header - the Service Index (SI) will indicate if a PNF(s) was at the end of the chain. The LSN changes the SPI in order that the underlay routes the metadata back directly to the KPI database (KPIDB).

Network Node/Element: Device that forwards packets or frames based on outer header information. In most cases is not aware of the presence of NSH.

Network Overlay: Logical network built on top of existing network (the underlay). Packets are encapsulated or tunneled to create the overlay network topology.

Network Service Header: Data plane header added to frames/packets. The header contains information required for service chaining, as well as metadata added and consumed by network nodes and service elements.

NSH Proxy: Acts as a gateway: removes and inserts SH on behalf of a service function that is not NSH aware.

Service Classifier: Function that performs classification and imposes an NSH. Creates a service path. Non-initial (i.e. subsequent) classification can occur as needed and can alter, or create a new service path.

Service Function (SF): A function that is responsible for specific treatment of received packets. A service function can act at the network layer or other OSI layers. A service function can be virtual instance or be embedded in a physical network element. One of multiple service functions can be embedded in the same network element. Multiple instances of the service function can be enabled in the same administrative domain.

Service Function Chain (SFC): A service function chain defines an ordered set of service functions that must be applied to packets and/or frames selected as a result of classification. The implied

order may not be a linear progression as the architecture allows for nodes that copy to more than one branch. The term service chain is often used as shorthand for service function chain.

Service Function Path (SFP): The instantiation of a SFC in the network. Packets follow a service function path from a classifier through the requisite service functions.

Stamping Controller SC: The SC may be part of the service chaining application, SDN controller, NFVO or any MANO entity. For clarity we define the SC separately here as the central logic that decides what packets to stamp and how. The SC instructs the classifier on how to build the NSH header.

Stamp Control Plane (SCP): the control plane between the FSN and the SC.

Key Performance Indicator Database (KPIDB): external storage of Metadata for reporting, trend analysis etc.

2.3. Abbreviations

DEI	Drop Eligible Indicator
DSCP	Differentiated Services Code Point
FSN	First Stamping Node
KPI	Key Performance Indicator
KPIDB	Key Performance Indicator Database
LSN	Last Stamping Node
MD	Metadata
NFV	Network Function Virtualization
NFVI-PoP	NFV Infrastructure Point of Presence
NIC	Network Interface Card
NSH	Network Service Header
OAM	Operations, Administration, and Maintenance

PCP	Priority Code Point
PNF	Physical Network Function
PNFN	Physical Network Function Node
QoE	Quality of Experience
QoS	Quality of Service
QS	QoS Stamp
RSP	Rendered Service Path
SC	Stamping Controller
SCL	Service Classifier
SCP	Stamp Control Plane
SI	Service Index
SF	Service Function
SFC	Service Function Chain
SFN	Service Function Node
SFP	Service Function Path
SSI	Stamp Service Index
TC	Traffic Class
TS	Timestamp
VLAN	Virtual Local Area Network
VNF	Virtual Network Function
vSwitch	Virtual Switch

3. NSH KPI Stamping

A typical KPI stamping architecture is presented in Figure 1.

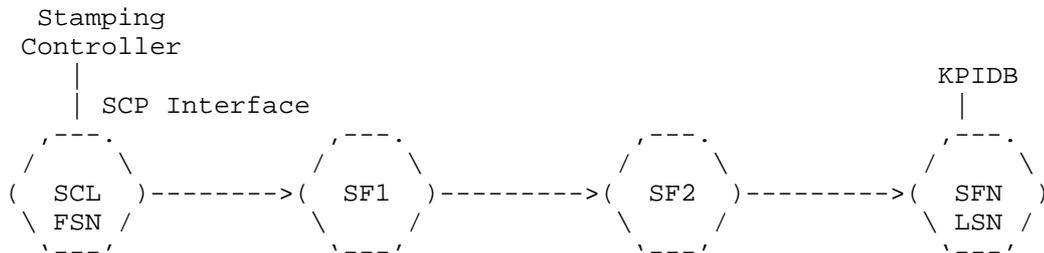


Figure 1: Logical roles in NSH KPI Stamping

The Stamping Controller (SC) will most probably be part of the SFC controller but is explained separately in this document for clarity. The SC is responsible for initiating start/stop stamp requests to the SCL or FSN, and also for distributing NSH stamping policy into the service chain via the Stamping Control Plane (SCP) interface.

The First Stamp Node (FSN) will typically be part of the SCL but again is called out as separate logical entity for clarity. The FSN is responsible for marking NSH MD fields for the correct flow with the appropriate NSH fields. This tells all upstream nodes how to behave in terms of stamping at VNF ingress, egress or both, or ignoring the stamp NSH metadata completely. The FSN also writes the Reference Time value, a (possibly inaccurate) estimate of the current time-of-day, into the header, allowing the {chain:flow} performance to be compared to previous samples for offline analysis. The FSN should return an error to the SC if not synchronized to the current time-of-day and forward the packet along the service-chain unchanged.

SF1, SF2 stamp the packets as dictated by the FSN and process the payload as per normal.

Note 1: The exact location of the stamp creation may not be in the VNF itself, as referenced in Section 3.3.

Note 2: Special cases exist where some of the SFs (PNFs or VNFs) are NSH-unaware. This is covered in Section 5.

The Last Stamp Node (LSN) should strip the entire header and forward the raw packet to the IP next hop. The LSN also exports NSH stamp information to the KPI Database (KPIDB) for offline analysis; the LSN may either export the stamping information of all packets, or a subset based on packet sampling. In fully virtualized environments the LSN will be co-located with the VNF that decrements the NSH

Service Index to zero. Corner cases exist whereby this is not the case and is covered in section 5.

3.1. Prerequisites

Timestamping presents a set of prerequisites not required to QoS-Stamp. In order to guarantee metadata accuracy, all servers hosting VNFs should be synchronized from a centralized stable clock. As it is assumed that PNFs do not timestamp there is no need for them to synchronize. There are two possible levels of synchronization:

Level A: Low accuracy time-of-day synchronization, based on NTP [RFC5905].

Level B: High accuracy synchronization (typically on the order of microseconds), based on [IEEE1588].

Each platform SHOULD have a level A synchronization, and MAY have a level B synchronization.

Level A requires each platform (including the Stamp Controller) to synchronize its system real-time-clock to an NTP server. This is used to mark the metadata in the chain, using the <Reference Time> field in the NSH KPIstamp header (Section 4.2). This timestamp is written to the NSH header by the first SF in the chain. NTP accuracy can vary by several milliseconds between locations. This is not an issue as the Reference Time is merely being used as a reference inserted into the KPIDB for performance monitoring.

Level B synchronization requires each platform to be synchronized to a Primary Reference Clock (PRC) using the Precision Time Protocol [IEEE1588]. A platform MAY also use Synchronous Ethernet ([G.8261], [G.8262], [G.8264]), allowing more accurate frequency synchronization.

If a SF is not synchronized at the moment of timestamping, it should indicate synch status in the NSH header. This is described in more detail in section 4.

By synchronizing the network in this way, the timestamping operation is independent of the current RSP, whether the entire chain is served by one NFVI-PoP or by multiple. Indeed the timestamp MD can indicate where a chain has been moved due to a resource starvation event as indicated in 0 below, between VNF 3 and VNF 4 at time B.

Referring to figure 3 above, x, v and y are notional sum values of the QoS configuration of the flow within a given chain. As the encapsulation of the flow can change from hop to hop in terms of VLAN header(s), MPLS labels, DSCP(s) these values are used to compare consistency of configuration from for example payload DSCP through overlay and underlay QoS settings in VLAN IEEE 802.1Q bits, TC MPLS bits and infrastructure DSCPs.

The above figure indicates that at VNF4 in the chain, the egress QoS marking is inconsistent. That is, the ingress QoS settings does not match the egress. The method described here will indicate which QoS field(s) is inconsistent, and whether this is ingress (whereby the underlay has incorrectly marked and queued the packet) or egress (where the VNF has incorrectly marked and queued the packet).

3.2. Operation

KPIstamping detection mode uses MD type 2. This involves the SFC classifier stamping the flow at chain ingress, and no subsequent stamps being applied, rather each VNF upstream can compare its local condition with the ingress value and take appropriate action. Therefore detection mode is very efficient in terms of header size that does not grow after the classification. This is further explained in section 4.1.

Section 3.5 of [NSH] (draft-ietf-sfc-nsh-10) defines NSH metadata type 2 encapsulation as per the figure below In KPIstamped detection and extended mode, flows will use this format.

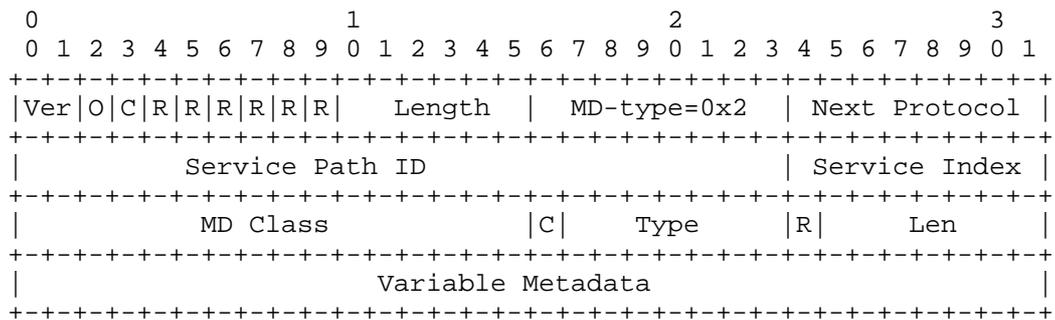


Figure 5: NSH MD type 2 Encapsulation

3.2.1. Flow Selection

The SC should maintain a list of flows within each service chain to be monitored. This flow table should be in the format SPI:5 tuple ID. The SC should map these pairs to unique Flow IDs per service chain within the extended NSH header specified in this draft. The SC should instruct the FSN to initiate timestamping on flow table match. The SC may also tell the classifier the duration of the timestamping operation, either by a number of packets in the flow or by a time duration.

In this way the system can monitor the performance of the all en-route traffic, or an individual subscriber in a chain, or just a specific application or QoS class the subscriber is running.

The SC should write the list of monitored flows into the KPIDB for correlation of performance and configuration data. Thus, when the KPIDB receives data from the LSN it understands to which flow the data pertains.

The association of source IP to subscriber identity is outside the scope of this draft and will vary by network application. For example, the method of association of a source IP to IMSI in mobile cores will be different to how a CPE with NAT function may be chained in an enterprise NFV application.

3.2.2. SCP Interface

A new Stamp control plane (SCP) interface is required between the SC and the FSN or classifier. This interface:

- o Queries the SFC classifier for a list of active chains and flows
- o Communicates which chains and flows to stamp. This can be a specific {chain:flow} combination or include wildcards for monitoring subscribers across multiple chains or multiple flows within one chain.
- o How the stamp should be applied (ingress, egress, both or specific).

- o Typically SCP timestamps flows for a certain duration for trend analysis, but only stamps one packet of each QoS class in a chain periodically (perhaps once per day or after a network change). Therefore timestamping is generally applied to a much larger set of packets than QoS stamping
- o When to stop stamping, either after a certain number of packets or duration.

Exact specification of SCP is for further study.

3.3. Performance Considerations

This draft does not mandate a specific stamping implementation method, and thus NSH KPI stamping can either be performed by hardware mechanisms, or by software. If software-based stamping is used, applying and operating on the stamps themselves incur an additional small delay in the service chain. However, it can be assumed that these additional delays are all relative for the flow in question. This is only pertinent for timestamping mode, and not for QoS stamping mode. Thus, whilst the absolute timestamps may not be fully accurate for normal non-timestamped traffic they can be assumed to be relative.

It is assumed that the method described in this document would only operate on a small percentage of user flows. The service provider may choose a flexible policy in the SC to timestamp a selection of user-plane every minute for example to highlight any performance issues. Alternatively, the LSN may selectively export a subset of the KPIstamps it receives, based on a predefined sampling method. Of course the SC can stress test an individual flow or chain should a deeper analysis be required. We can expect that this type of deep analysis has an impact on the performance of the chain itself whilst under investigation. The impact will be dependent on vendor implementation and outside the scope of this document.

For QoS stamping the method described here is even less intrusive, as you would not typically need to QoS stamp multiple packets in a flow rather periodically (perhaps once per day) check one packet in a chain per QoS class.

The KPIstamp may be applied at various parts of the NFV architecture. The VNF, hypervisor, vSwitch or NIC are all potential locations that can append the packet with the requested KPIstamp. Whilst it is desirable to stamp as close as possible to the VNF for accuracy, the exact location of the stamp application is outside the scope of this document, but should be consistent across the individual SC domain.

4. NSH KPIStamping Encapsulation

KPI stamping uses NSH MD type 0x02 for detection of anomalies and extended mode for root cause analysis of KPI violations. These are further explained in this section.

4.1. KPIStamping Encapsulation (Detection Mode)

The generic NSH MD type 2 allocation for KPI Stamping (detection mode) is shown below. This is the format we propose for KPI anomaly detection.

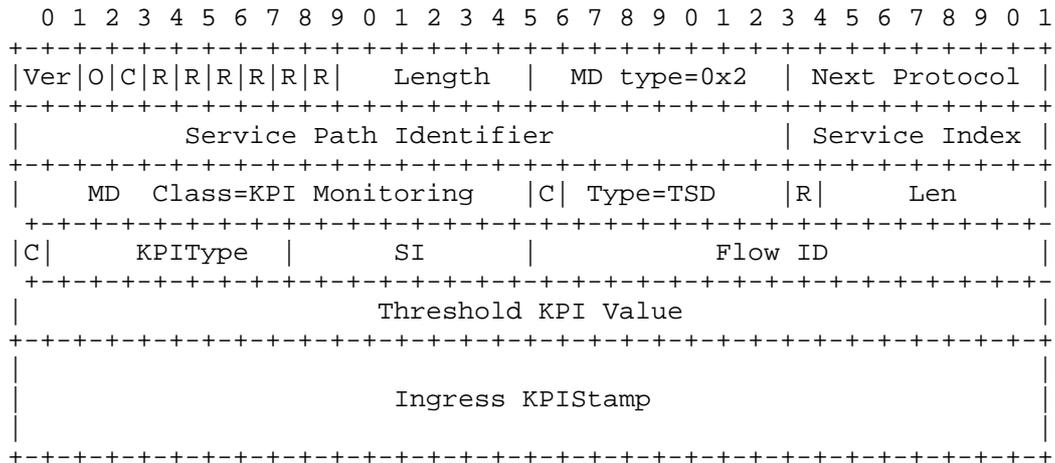


Figure 6: Generic NSH KPI Encapsulation (Detection Mode)

Relevant fields in header that the FSN must implement:

- o The O bit should not be set as we are operating on subscriber packets
- o The C bit should be set indicating critical metadata exists
- o The MD type must be set to 0x2
- o The MD Class must be set to 0x10 (General KPI Monitoring) as requested in Section 9. The stamp type is defined as per below:
 - o Type = 0x00 Reserved.
 - o Type = 0x01 Timestamp Detection
- o The MSB of the Type field must be set to zero. Thus if a receiver along the path does not understand the KPIstamping protocol it will pass the packet transparently and not drop. This scheme allows for extensibility to the mechanism described in this document to other KPI collections and operations.

In the first header the SFC classifier may program a KPI threshold value. This is a value that when exceeded, requires the SF to set the C bit and insert the current SI value into the SI field. The KPI type is the type of KPI stamp inserted into the header as per section 9.

The flow ID is inserted into the header by the SFC classifier in order to correlate flow data in the KPIDB for offline analysis. The last two mandatory context headers are reserved for the KPIStamp. This is the KPI value at the chain ingress at the SFC classifier.

As an example operation, say we are using KPI type 0x01 (timestamp) when a service function (SFn) receives the packet it can compare current local timestamp (it first checks that it is synchronized to network PRC) with chain ingress timestamp to calculate the latency in the chain. If this value exceeds the timestamp threshold, it then sets the C bit inserts its SI and returns the NSH header to the KPIDB. This effectively tells the system that at SFn the packet violated the KPI threshold. All subsequent upstream SFs perform no NSH KPI operation as the flow has already been marked in violation via the C bit. Please refer to figure 9 for timestamp format.

When this occurs the SFC control plane system would then invoke the KPI extended mode, which uses a more sophisticated (and intrusive) method to isolate KPI violation root cause as described below.

Note: Whilst detection mode is a valuable tool for latency actions, we feel that it is not justified to build the logic into the KPI

system for QoS configuration. As QoS stamping is done infrequently and on a tiny percentage of user plane, it is more practical to use extended mode only for service chain QoS verification.

The generic NSH MD type 2 KPI Stamping header extended mode is shown below. This is the format we propose for performance monitoring of service chain issues with respect to QoS configuration and latency.

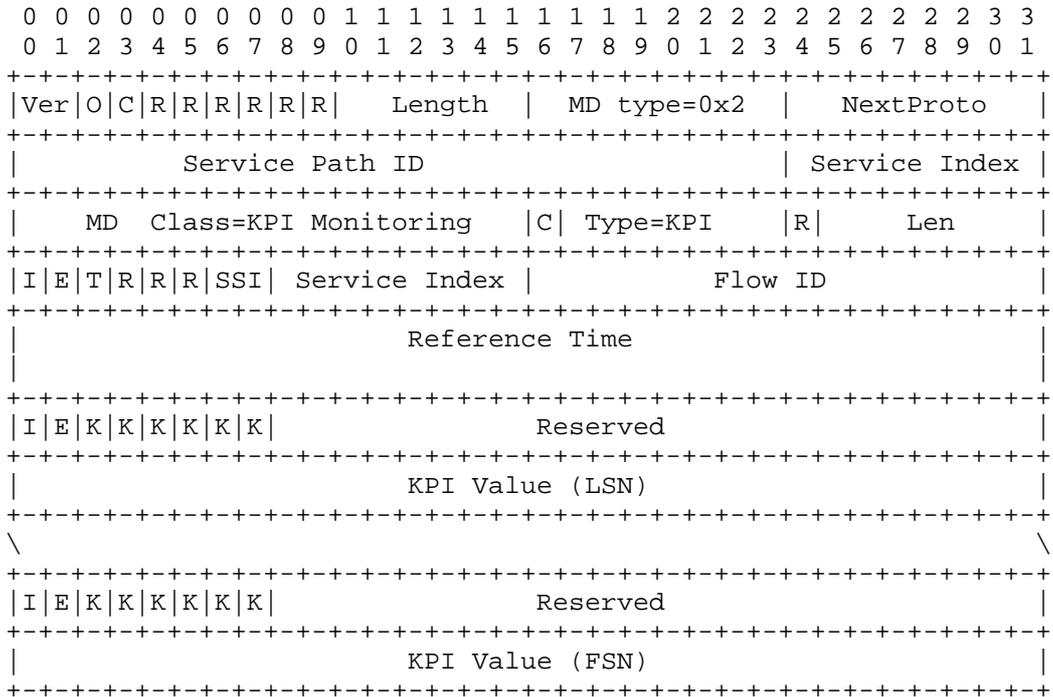


Figure 7: Generic KPI Encapsulation (Extended Mode)

As per section 9, we propose a new MD class 0x10 to indicate KPI MD. Within this class we define 2 types for QoS and timestamp MD to be reported along the service chain. The K bits are KPI specific bits, for example, SYN for timestamping.

4.2. NSH Timestamping Encapsulation (Extended Mode)

The NSH timestamping encapsulation is shown below.

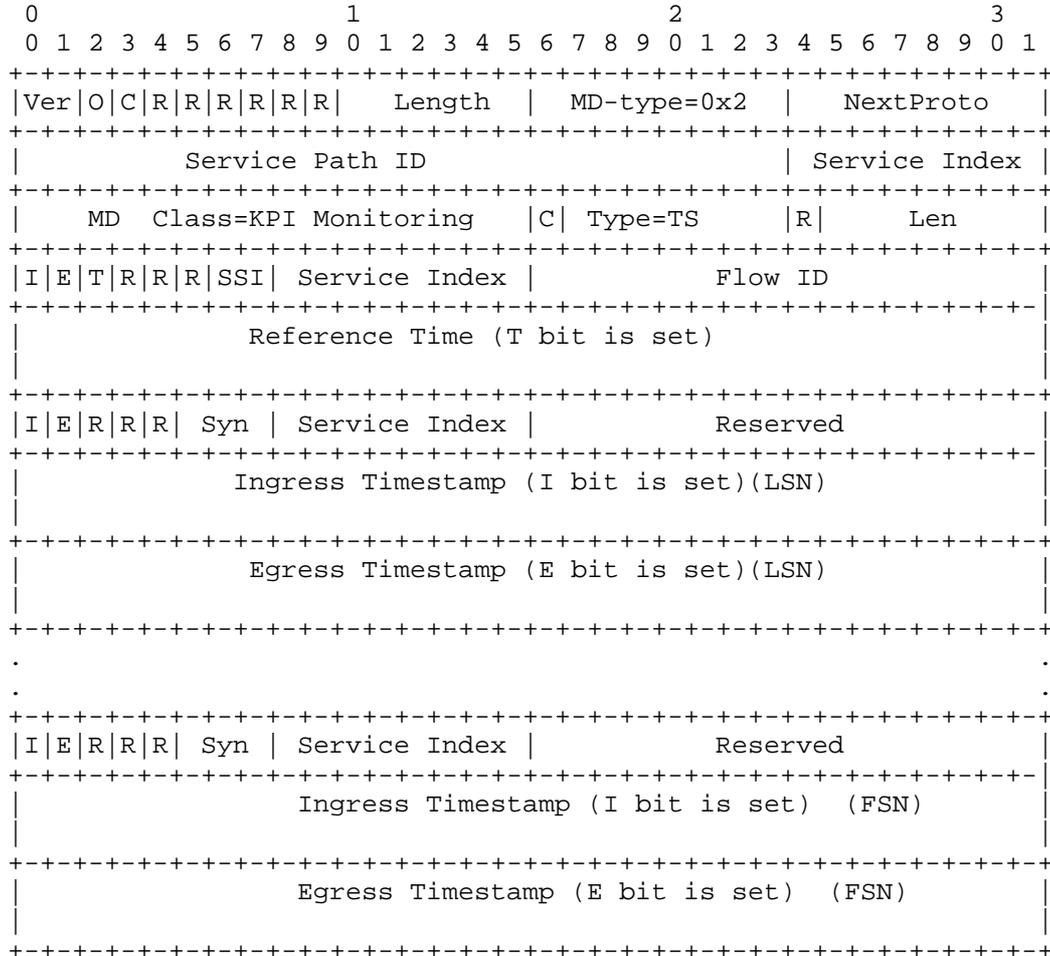


Figure 8: NSH Timestamp Encapsulation (Extended Mode)

Relevant fields in header that the FSN must implement:

- o The O bit should not be set as we are operating on subscriber packets
- o The C bit should be set indicating critical metadata exists
- o The MD type must be set to 0x2
- o The MD Class must be set to 0x10 (General KPI Monitoring) as requested in Section 9. The stamp type is defined as per below:
 - o Type = 0x00 Reserved.
 - o Type = 0x02 Timestamp Extended
 - o Type = 0x03 QoSStamp Extended
- o The MSB of the Type field must be set to zero. Thus if a receiver along the path does not understand the KPIstamping protocol it will pass the packet transparently and not drop. This scheme allows for extensibility to the mechanism described in this document to other KPI collections and operations.

The FSN KPIstamp metadata starts with Stamping Configuration Header. This header contains the Stamp Service Index (SSI) field which must be set to one of the following values:

- o 0x0 KPIstamp mode, no Service index specified in the Stamp Service Index field.
- o 0x1 KPIUstamp Hybrid mode is selected, Stamp Service Index contains LSN Service index. This is used when PNFs or NSH-unaware SFs are used at the tail of the chain. If SSI=0x1, then the value in the type field informs the chain which SF should act as the LSN.
- o 0x2 KPIstamp Specific mode is selected, Stamp Service Index contains the targeted Service Index. In this case the Stamp Service Index field indicates which SF is to be stamped. Both ingress and egress stamps are performed when the SI=SSI on the chain. For timestamping mode, the FSN will also apply the Reference Time and Ingress Timestamp. This will indicate the delay along the entire service chain to the targeted SF. This method may also be used as a light implementation to monitor end-to-end service chain performance whereby the targeted SF is the LSN. This is not applicable to QoSStamping mode.

The Flow ID is a unique 16 bit identifier written into the header by the classifier. This allow 65536 flows to be concurrently stamped on any given NSH service chain (SPI). Flow IDs are not written by subsequent SFs in the chain. The FSN may export monitored flow IDs to the KPIDB for correlation.

The E bit should be set if Egress stamp is requested.

The I bit should be set if Ingress stamp is requested.

The T bit should be set if Reference Time follows Stamping Configuration Header.

Reference Time is the wall clock of the FSN, and may be used for historical comparison of SC performance. If the FSN is not Level A synchronized (see Section 3.1) it should inform the SC over the SCP interface. The Reference Time is represented in 64-bit NTP format [RFC5905].

Each stamping Node adds stamping metadata which consist of Stamping Reporting Header and timestamps.

The E bit should be set if Egress stamp is reported.

The I bit should be set if Ingress stamp is reported.

With respect to timestamping mode, the Syn bits are an indication of the synchronization status of the node performing the timestamp and must be set to one of the following values:

- o In Synch: 0x00
- o In holdover: 0x01
- o In free run: 0x02
- o Out of Synch: 0x03

If the network node is out of synch or in free run no timestamp is applied by the node (but other timestamp MD is applied) and the packet is processed normally.

If FSN is out of synch or in free run timestamp request rejected and not propagated though the chain. The FSN should inform the SC in such an event over the SCP interface.

The outer service index value is copied into the stamp metadata to help cater for hybrid chains that's are a mix of VNFs and PNFs or through SFs that do not understand NSH. Thus if a flow transits through a PNF or an NSH-unaware node the delta in the inner service index between timestamps will indicate this.

The Ingress Timestamp and Egress Timestamp are represented in 64-bit NTP format [RFC5905]. The corresponding bits (I and E) reported in the Stamping Reporting Header of the node's metadata.

The 64-bit timestamp format [RFC5905] is presented below:

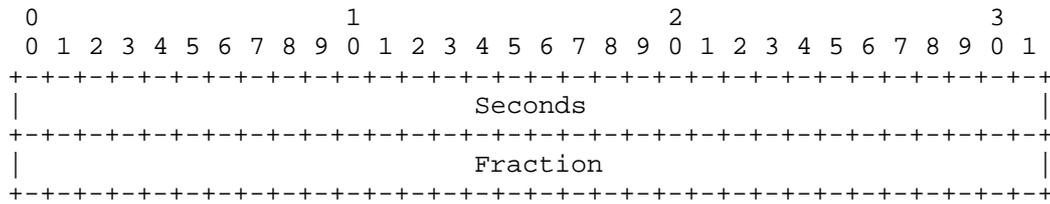


Figure 9: NTP [RFC5905] 64-bit Timestamp Format

4.3. NSH QoS Stamping Encapsulation (Extended Mode)

Packets have a variable QoS stack. That is for example the same payload IP can have a very different stack in the access part of the network to the core. This is most apparent in mobile networks where for example in an access circuit we would have 2 layers of infrastructure IP header (DSCP) - one transport-based and the other IPsec-based, in addition to multiple MPLS and VLAN tags. The same packet as it leaves the PGW Gi egress interface may be very much simplified in terms of overhead and related QoS fields.

Because of this variability we need to build extra meaning into the QoS headers - they are not for example all PTP timestamps of a fixed length as in the case of timestamping, rather they are variable lengths and types. Also they can be changed on the underlay at any time without knowledge by the SFC system. Therefore each VNF must be able to ascertain and record its ingress and egress QoS configuration on the fly.

The suggested QoS type, lengths are as below. The type is 4 bits long.

Q Type(QT)	Value	Length	Comment
IVLAN	0x01	4 Bits	Ingress VLAN (PCP + DEI)
EVLAN	0x02	4 Bits	Egress VLAN
IQINQ	0x03	8 Bits	Ingress QinQ (2x PCP+DEI)
EQINQ	0x04	8 Bits	Egress QinQ
IMPLS	0x05	3 Bits	Ingress Label
EMPLS	0x06	3 Bits	Egress Label
IMPLS	0x07	6 Bits	2 Ingress Labels (2x EXP)
EMPLS	0x08	6 Bits	2 Egress Labels
IDSCP	0x09	8 Bits	Ingress DSCP
EDSCP	0x0A	8 Bits	Egress DSCP

For stacked headers such as MPLS and 802.1ad, we extract the QoS relevant data from the header and insert into one QoS value in order to be more efficient on packet size. This for MPLS we represent both EXP fields in one QoS value, and both 802.1p priority and drop precedence in one QoS value as indicated above.

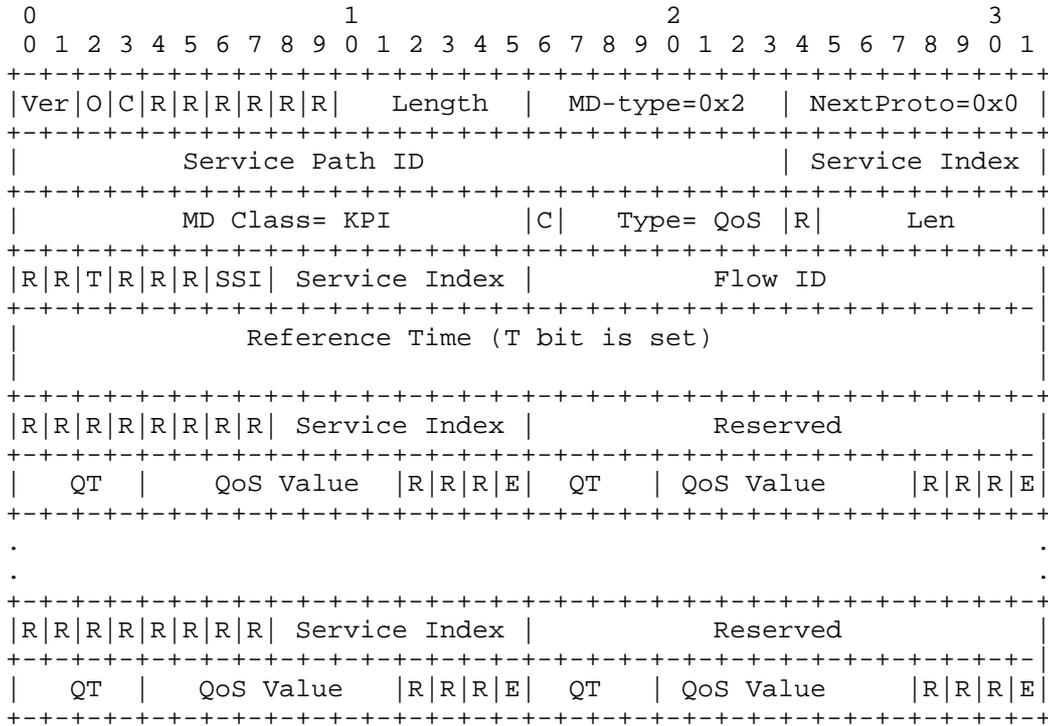


Figure 10: NSH QoS Configuration Encapsulation (Extended Mode)

The encapsulation above is very similar to that detailed in section 4.1 with the following exceptions

- I and E bits are not required as we wish to walk the full QoS stack at ingress and egress at every SF.
- Syn status bits are not required
- The QT (QoS Type) and QoS value are as outlined in the table above
- The E bit at the tail of each QoS context field indicates if this is the last egress QoS stamp for a given SF. This should coincide with SI=0 at the LSN, whereby the packet is truncated and the NSH MD sent to the KPIDB and the subscriber raw IP packet forwarded to the underlay next hop.

Note: It is possible to compress the frame structure to better utilize the header, but this would come at the expense of crossing byte boundaries. For ease of implementation, and that QoS stamping is applied on an extremely small subset of user plane traffic, we believe the above structure is a pragmatic compromise between header efficiency and ease of implementation.

5. Hybrid Models

A hybrid chain may be defined as a chain whereby there is a mix of NSH-aware and NSH-unaware SFs. This may be the case if some PNFs are used in the chain or if VNFs are used that do not support NSH.

Example 1. PNF in the middle

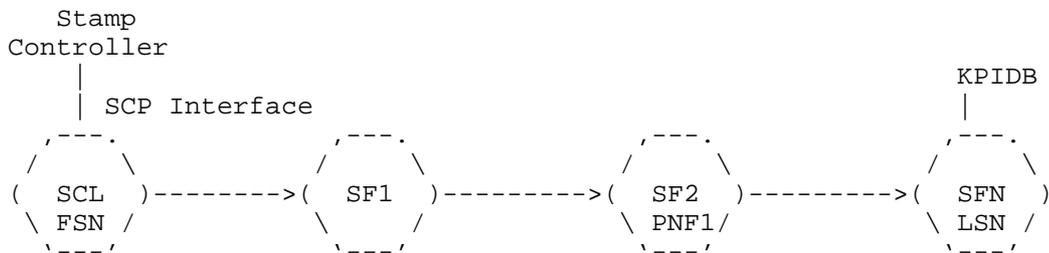


Figure 11: Hybrid chain with PNF in middle

In this example the FSN begins operation and sets the SI to 3, SF1 decrements this to 2 and passes the flow to an SFC proxy (not shown).

The proxy strips the NSH header and passes to the PNF. On receipt back from the PNF the Proxy decrements the SI and passes the packet onto the LSN with a SI=1.

After the LSN processes the traffic it knows it is the last node on the chain from the SI value and exports the entire NSH header and all metadata to the KPIDB. The payload is forwarded to the next hop on the underlay minus the NSH header. The TS information packet may be given a new SPI to act as a homing tag to transport the timestamp data back to the KPIDB.

Example 2. PNF at the end

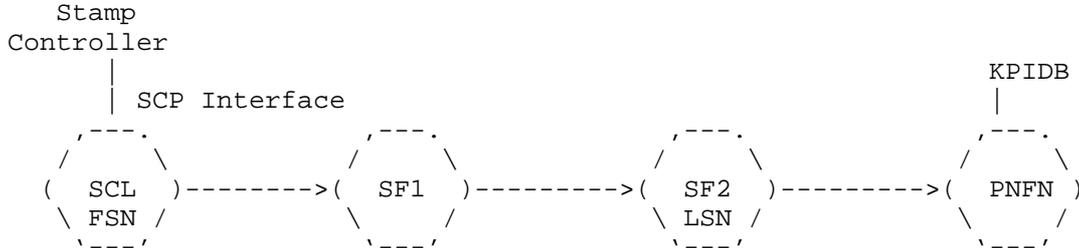


Figure 12: Hybrid Chain with PNF at end

In this example the FSN begins operation and sets the SI to 3, the SSI field set to 0x1, and the type to 1. Thus when SF2 receives the packet with SI=1, it understands that it is expected to take on the role of the LSN as it is the last NSH-aware node in the chain.

5.1. Targeted VNF Stamp

For the majority of flows within the service chain, stamps (ingress, egress or both) will be carried out at each hop until the SI decrements to zero and the NSH header and Stamp MD is exported to the KPIDB. There may exist however the need to just test a particular VNF (perhaps after a scale out operation, software upgrade or underlay change for example). In this case the FSN should mark the NSH header as follows:

SSI field is set to 0x2. Type is set to the expected SI at the SF in question. When outer SI is equal to the SSI, stamps are applied at SF ingress and egress, and the NSH header and MD are exported to the KPIDB.

6. Fragmentation Considerations

The method described in this draft does not support fragmentation. The SC should return an error should a stamping request from an external system exceed MTU limits and require fragmentation.

Depending on the length of the payload and the type of KPIstamp and chain length, this will vary for each packet.

In most service provider architectures we would expect a SI \ll 10, and that may include some PNFs in the chain which do not add overhead. Thus for typical IMIX packet sizes we expect to be able to perform timestamping on the vast majority of flows without fragmenting. Thus the classifier can have a simple rule to only allow KPIstamping on packet sizes less than 1200 bytes for example.

7. Security Considerations

The security considerations of NSH in general are discussed in [NSH].

The use of in-band timestamping, as defined in this document, can be used as a means for network reconnaissance. By passively eavesdropping to timestamped traffic, an attacker can gather information about network delays and performance bottlenecks.

The NSH timestamp is intended to be used by various applications to monitor the network performance and to detect anomalies. Thus, a man-in-the-middle attacker can maliciously modify timestamps in order to attack applications that use the timestamp values. For example, an attacker could manipulate the SFC classifier operation, such that it forwards traffic through 'better' behaving chains. Furthermore, if timestamping is performed on a fraction of the traffic, an attacker can selectively induce synthetic delay only to timestamped packets, causing systematic error in the measurements.

Similarly, if an attacker can modify QoS stamps, erroneous values may be imported into the KPIDB, resulting in further misconfiguration and subscriber QoE impairment.

An attacker that gains access to the SCP can enable time and QoS stamping for all subscriber flows, thereby causing performance bottlenecks, fragmentation, or outages.

As discussed in previous sections, NSH timestamping relies on an underlying time synchronization protocol. Thus, by attacking the time protocol an attack can potentially compromise the integrity of the NSH timestamp. A detailed discussion about the threats against time protocols and how to mitigate them is presented in [RFC7384].

8. Open Items for WG Discussion

- o Specification and operation of SCP

- o AOB

9. IANA Considerations

MD Class Allocation

MD classes are defined in [NSH].

IANA is requested allocate a new MD class value:

0x10 KPI General Monitoring, stamping types and QoS types.

NSH Stamping MD Types

IANA is requested to set up a registry of "NSH KPIstamping MD Types". These are 7-bit values. Registry entries are assigned by using the "IETF Review" policy defined in [RFC5226].

IANA is requested to allocate two new types as follows:

- o Type = 0x00 Reserved.
- o Type = 0x01 Timestamp Detection
- o Type = 0x02 Timestamp Extended
- o Type - 0x03 QoSStamp Extended

QoS Types (QT)

- o IVLAN 0x01
- o EVLAN 0x02
- o IQINQ 0x03
- o EQINQ 0x04
- o IMPLS 0x05
- o EMPLS 0x06
- o IMPLS 0x07
- o EMPLS 0x08

- o IDSCP 0x09
- o EDSCP 0x0A

10. Contributors

This document originated as draft-browne-sfc-nsh-timestamp-00 and had the following co-authors and contributors. We would like to thank and recognize them and their contributions.

Yoram Moses

Technion

moses@ee.technion.ac.il

Brendan Ryan

Intel Corporation

brendan.ryan@intel.com

11. Acknowledgments

This document was prepared using 2-Word-v2.0.template.dot.

The authors would like to thank Ramki Krishnan and Anoop Ghanwani from Dell for their reviews and comments on this draft.

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [NSH] Quinn, P., Elzur, U., "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.

12.2. Informative References

- [IEEE1588] IEEE TC 9 Instrumentation and Measurement Society, "1588 IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems Version 2", IEEE Standard, 2008.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.
- [RFC5905] Mills, D., Martin, J., Burbank, J., Kasch, W., "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.
- [RFC7384] Mizrahi, T., "Security Requirements of Time Protocols in Packet Switched Networks", RFC 7384, October 2014.
- [Y.1731] ITU-T Recommendation G.8013/Y.1731, "OAM Functions and Mechanisms for Ethernet-based Networks", August 2015.
- [Y.1564] ITU-T Recommendation Y.1564, "Ethernet service activation test methodology", March 2011.
- [G.8261] ITU-T Recommendation G.8261/Y.1361, "Timing and synchronization aspects in packet networks", August 2013.
- [G.8262] ITU-T Recommendation G.8262/Y.1362, "Timing characteristics of a synchronous Ethernet equipment slave clock", January 2015.
- [G.8264] ITU-T Recommendation G.8264/Y.1364, "Distribution of timing information through packet networks", May 2014.

Authors' Addresses

Rory Browne
Intel
Dromore House
Shannon
Co.Clare
Ireland

Email: rory.browne@intel.com

Andrey Chilikin
Intel
Dromore House
Shannon
Co.Clare
Ireland

Email: andrey.chilikin@intel.com

Tal Mizrahi
Marvell
6 Hamada St.
Yokneam, 20692 Israel

Email: talmi@marvell.com

Service Function Chaining (sfc)
Internet-Draft
Intended status: Informational
Expires: April 26, 2017

M. Boucadair, Ed.
Orange
October 23, 2016

Service Function Chaining (SFC) Control Plane Components & Requirements
draft-ietf-sfc-control-plane-08

Abstract

This document describes requirements for conveying information between Service Function Chaining (SFC) control elements and SFC data plane functional elements. Also, this document identifies a set of control interfaces to interact with SFC-aware elements to establish, maintain or recover service function chains. This document does not specify protocols nor extensions to existing protocols.

This document exclusively focuses on SFC deployments that are under the responsibility of a single administrative entity. Inter-domain considerations are out of scope.

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 April 26, 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	3
1.1.	Scope	3
1.2.	Terminology	4
1.3.	Assumptions	5
2.	Generic Considerations	6
2.1.	Generic Requirements	6
2.2.	SFC Control Plane Bootstrapping	6
2.3.	SFC Dynamics	7
2.4.	Coherent Setup of an SFC-enabled Domain	8
3.	SFC Control Plane: Reference Architecture & Interfaces	8
3.1.	Reference Architecture	8
3.2.	Centralized vs. Distributed	10
3.3.	Interface Reference Points	11
3.3.1.	C1: Interface between SFC Control Plane & SFC Classifier	11
3.3.2.	C2: Interface between SFC Control Plane & SFF	13
3.3.3.	C3: Interface between SFC Control Plane & SFC-aware SFs	14
3.3.4.	C4: Interface between SFC Control Plane & SFC Proxy	16
4.	Additional Considerations	16
4.1.	Discovery of the SFC Control Element	16
4.2.	SF Symmetry	17
4.3.	Pre-deploying SFCs	17
4.4.	Withdraw a Service Function (SF)	17
4.5.	SFC/SFP Operations	18
4.6.	Unsolicited (Notification) Messages	18
4.7.	Liveness Detection	18
4.8.	Monitoring & Counters	19
4.9.	Validity Lifetime	19
4.10.	Considerations Specific to the Centralized Path Computation Model	20
4.10.1.	Service Function Path Adjustment	20
4.10.2.	Head End Initiated SFP Establishment	21
4.10.3.	(Regional) Restoration of Service Functions	21
4.10.4.	Fully Controlled SFF/SF Sequence for a SFP	22
5.	Security Considerations	23
5.1.	Secure Communications	23
5.2.	Pervasive Monitoring	24
5.3.	Privacy	24
5.4.	Denial-of-Service (DoS)	24

5.5. Illegitimate Discovery of SFs and SFC Control Elements	24
6. IANA Considerations	24
7. Acknowledgments	24
8. Contributors	25
9. References	27
9.1. Normative References	27
9.2. Informative References	27
Author's Address	29

1. Introduction

The dynamic enforcement of a service-derived forwarding policy for packets entering a network that supports advanced Service Functions (SFs) has become a key challenge for operators. Typically, many advanced Service Functions (e.g., Performance Enhancement Proxies ([RFC3135]), NATs [RFC3022][RFC6333][RFC6146], firewalls [I-D.ietf-opsawg-firewalls], etc.) are solicited for the delivery of value-added services, particularly to meet various service objectives such as IP address sharing, avoiding covert channels, detecting and protecting against ever increasing Denial-of-Service (DoS) attacks, etc.

Because of the proliferation of such advanced service functions together with complex service deployment constraints that demand more agile service delivery procedures, operators need to rationalize their service delivery logics and master their complexity while optimising service activation time cycles. The overall problem space is described in [RFC7498]. A more in-depth discussion on use cases can be found in [I-D.ietf-sfc-use-case-mobility] and [I-D.ietf-sfc-dc-use-cases].

[RFC7665] presents a model addressing the problematic aspects of existing service deployments, including topological dependence and configuration complexity. It also describes an architecture for the specification, creation, and ongoing maintenance of Service Function Chains (SFC) within a network. That is, how to define an ordered set of Service Functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification. [I-D.ietf-sfc-nsh] specifies the SFC encapsulation as per [RFC7665].

1.1. Scope

While [RFC7665] focuses on data plane considerations, this document describes requirements for conveying information between SFC control elements and SFC data plane functional elements. Also, this document identifies a set of control interfaces to interact with SFC-aware elements to establish, maintain or recover service function chains.

Both distributed and centralized control plane schemes to install SFC-related state and influence forwarding policies are discussed.

This document does not make any assumption on the deployment use cases. In particular, the document implicitly covers fixed, mobile, data center networks, and any combination thereof.

This document does not make any assumption about which control protocol to use, whether one or multiple control protocols are required, or whether the same or distinct control protocols will be invoked for each of the control interfaces. It is out of scope of this document to specify a profile for an existing protocol, to define protocol extensions, or to select a protocol.

Considerations related to the chaining of Service Functions (SFs) that span domains owned by multiple administrative entities are out of scope.

It is out of scope of this document to discuss SF-specific control and policy enforcement schemes; only SFC considerations are elaborated, regardless of the various connectivity services that may be supported in the SFC-enabled domain. Likewise, only the control of SFC-aware elements is discussed.

Service catalogue (including guidelines for deriving service function chains) is out of scope.

This document does not specify any flow exchange to illustrate the comprehensive SFC operation. Instead, it focuses on the required information to be conveyed via each control interface. Note that sketching a comprehensive flow exchange is also a function of deployment considerations that are out of scope.

1.2. Terminology

The reader should be familiar with the terms defined in [RFC7498] and [RFC7665].

The document makes use of the following terms:

- o SFC data plane functional element: Refers to SFC-aware Service Function, Service Function Forwarder (SFF), SFC proxy, or classifier as defined in the SFC data plane architecture [RFC7665].
- o SFC Control Element: A logical entity that instructs one or more SFC data plane functional elements on how to process packets within an SFC-enabled domain.

- o SFC Classification rule: Refers to a rule maintained by a classifier that reflects the policies for binding an incoming flow/packet to a given SFC and Service Function Path (SFP). Actions are associated with matching criteria. The set of classification entries maintained by a classifier are referred to as in the classification policy table.
- o SFP Forwarding Policy Table: this table reflects the SFP-specific traffic forwarding policy enforced by SFF components for every relevant incoming packet that is associated to one of the existing SFCs. The SFP Identifier (SFP-id) is used as a lookup key to determine forwarding action regardless of whether the SFC is fully constrained, partially constrained, or not constrained at all. Additional information such as a flow identifier, Service Index (SI), and/or other characteristics (e.g., the 5-tuple transport coordinates of the original packet) may be used for lookup purposes. The set of information to use for lookup purposes may be instructed by the control plane.

1.3. Assumptions

This document adheres to the assumptions listed in Section 1.2 of [RFC7665].

As a reminder, a Service Function Path (SFP) designates a subset of the collection designated by the SFC. For some SFPs, in some deployments, that will be a set of 1. For other SFPs (in the same or other deployments) it may be a larger set. For some SFPs in some deployments the SFP may designate the same set of choices as the SFC. This document accommodates all those deployments.

This document does not make any assumptions about the co-location of SFC data plane functional elements; this is deployment-specific. This document can accommodate a variety of deployment contexts such as (but not limited to):

- o A Service Function Forwarder (SFF) can connect instances of the same or distinct SFs.
- o An SF instance can be serviced by one or multiple SFFs.
- o One or multiple SFs can be co-located with an SFF.
- o A boundary node (that connects one SFC-enabled domain to a node either located in another SFC-enabled domain or in a domain that is SFC-unaware) can act as an egress node and an ingress node for the same flow.
- o Distinct ingress and egress nodes may be crossed by a packet when forwarded in an SFC-enabled domain.
- o Distinct ingress nodes may be solicited for each traffic direction (e.g., upstream and downstream).

- o The same boundary node may act as an ingress node, an egress node, and also embed a classifier.
- o A classifier can be hosted in a node that embeds one or more SFs.
- o Many network elements within an SFC-enabled domain may behave as egress/ingress nodes.

Furthermore, the following assumptions are made:

- o A Control Element can be co-located with a classifier, SFF or SF.
- o One or multiple Control Elements can be deployed in an SFC-enabled domain.
- o State synchronization between Control Elements is out of scope.

2. Generic Considerations

2.1. Generic Requirements

Some deployments require that forwarding within an SFC-enabled domain must be allowed even if no control protocols are enabled. Static configuration must be allowed.

A permanent association between an SFC data plane element with a Control Element must not be required; specifically, the SFC-enabled domain must keep on processing incoming packets according to the SFC instructions even during temporary unavailability events of control plane components. SFC implementations that do not meet this requirement will suffer from another flavor of the constrained high availability issue, discussed in Section 2.3 of [RFC7498], supposed to be solved by SFC designs.

2.2. SFC Control Plane Bootstrapping

The interface that is used to feed the SFC control plane with service objectives and guidelines is not part of the SFC control plane itself. Therefore, this document assumes the SFC control plane is provided with a set of required information for proper SFC operation with no specific assumption about how this information is collected/provisioned, nor about the structure of such information. The following information that is recommended to be provided to the SFC control plane prior to bootstrapping includes:

- o Locators for classifiers/SFF/SFs/SFC proxies, etc.
- o SFs serviced by each SFF.
- o A list of service function chains, including how they are structured and unambiguously identified.
- o Status of each SFC: active/pre-deployment phase/etc. An SFC can be defined at the management level and instantiated in an SFC-enabled domain for pre-deployment purposes (e.g., testing).

Actions to activate, modify or withdraw an SFC are triggered by the control plane. Nevertheless, this document does not make any assumption about how an operator instructs the control plane.

- o A list of classification guidelines and/or rules to bind flows to SFCs/SFPs.
- o Security credentials.
- o Context information that needs to be shared on a per SFC basis.

Optionally, load balancing objectives at the SFC level or on a per node (e.g., per-SF/SFF/SFC proxy) basis may also be provided to the SFC control plane. Likewise, the set of metadata that is supported by SFC-aware SFs, SFFs, and SFC proxies may be provided to the SFC control plane.

Also, the SFC control plane may gather the following information from an SFC-enabled domain at bootstrapping (non-exhaustive list). How this information is collected is left unspecified in this document:

- o The list of active SFC-aware SFs (including their locators).
- o The list of SFFs and the SFs that are attached to.
- o The list of enabled SFC proxies, and the list of SFC-unaware SFs attached to.
- o The list of active SFCs/SFPs as enabled in an SFC-enabled domain.
- o The list of classifiers and their locators, so as to retrieve the classification policy table for each classifier, in particular.
- o The SFP Forwarding Policy Tables maintained by SFFs.
- o The set of metadata that is supported by SFC-aware SFs, SFFs, and SFC proxies. Additional capabilities (e.g., supported transport encapsulation scheme(s), supported SFC header version(s)) may also be collected.

During the bootstrapping phase, a Control Element may detect a conflict between the running configuration in an SFC data plane element and the information maintained by the control plane. Consequently, the control plane undertakes appropriate actions to fix those conflicts. This is typically achieved by invoking one of the interfaces defined in Section 3.3.

After bootstrapping, the SFC control plane is fed (dynamically or on a per request basis) with a set of information that is required for proper SFC operation. More details about this information are discussed in Section 3 and Section 4.

2.3. SFC Dynamics

By default, SFC data and control plane elements must assume that SFC control information are dynamic by nature. This requirement applies even for policies that are communicated via an upper layer to

communicate service objectives and guidelines to a control element. Additionally, the SFC control plane must not assume that the capabilities of SFC data plane elements are frozen. The SFC control architecture must be designed to accommodate any dynamic of SFs/SFFs attachments, software updates, dynamic network condition events, etc.

The overall SFC orchestration is not discussed in this document because SFC operations are likely to be policy-driven. Nevertheless, the document specifies required interfaces that can be invoked in the context of an SFC orchestration fed with policies that are local to an SFC-enabled domain. No assumption is made about those policies nor their change dynamics. The control interfaces are designed to cover both dynamic control information exchange, but also to issue request solicitations to the appropriate SFC data plane elements.

2.4. Coherent Setup of an SFC-enabled Domain

Various transport encapsulation schemes and/or versions of SFC header implementations may be supported by one or several nodes of an SFC-enabled domain. For the sake of coherent configuration, the SFC control plane is responsible for instructing all the involved SFC data plane functional elements about the behavior to adopt to select the transport encapsulation scheme(s), the version of the SFC header to enable, etc.

3. SFC Control Plane: Reference Architecture & Interfaces

3.1. Reference Architecture

The SFC control plane is responsible for the following:

- o Build and monitor the service-aware topology. For example, this can be achieved by means of dynamic SF discovery techniques. Those means are out of scope of this document.
- o Maintain a repository of service function chains, SFC matching criteria to bind flows to a given service function chain, and mapping between service function chains and SFPs.
- o Guarantee the coherency of the configuration and the operation of an SFC-enabled domain.
- o Dynamically compute a service forwarding path (distributed model, see Section 3.2).
- o Determine a forwarding path in the context of a centralized deployment model (see Section 3.2).
- o Update service function chains or adjust SFPs (e.g., for restoration purposes) based on various inputs (e.g., external policy context, path alteration, SF unavailability, SF withdrawal, service decommissioning, etc.).

- o Provision SFP Forwarding Policy Tables of involved SFFs and provide classifiers with traffic classification rules.

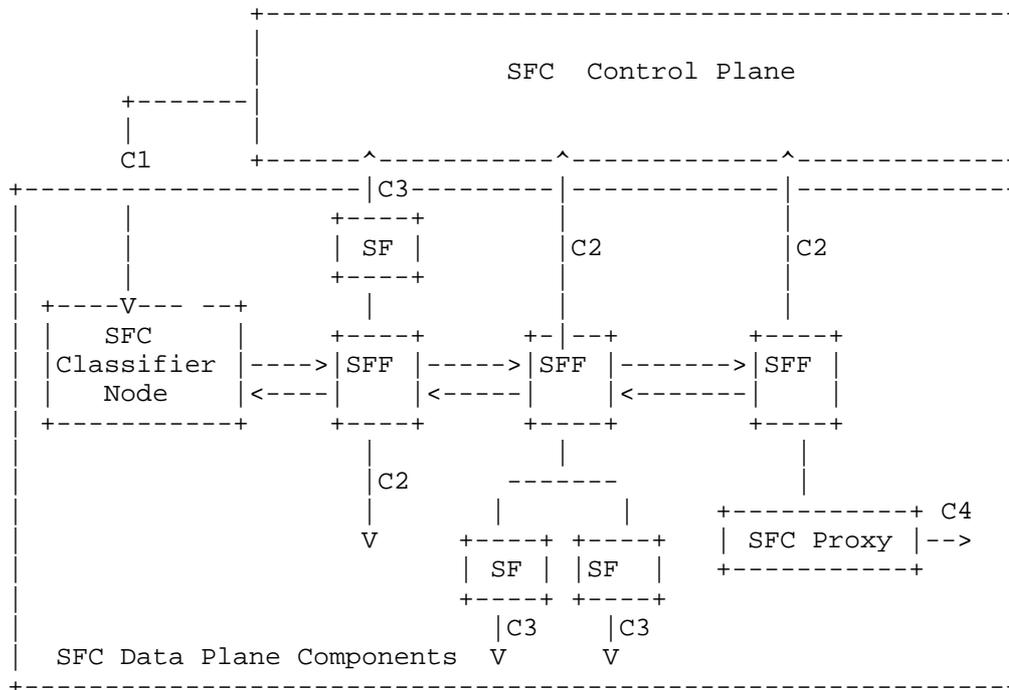


Figure 1: SFC Control Plane Interfaces

Figure 1 shows the overall SFC control plane architecture, including interface reference points. Particularly, Figure 1 shows the various interfaces that are required for conveying control information between the SFC control plane and underlying SFC data plane elements:

1. Interface between SFC Control Plane & SFC classifier (C1): This interface is used to manage SFC classification rules in classifiers. These rules can be added, modified, or deleted. Additional information is provided in Section 3.3.1. In order to avoid stale classification rules and to allow for local sanity checks, a validity lifetime is associated with each classification rule (Section 4.9).
2. Interface between SFC Control Plane & SFF (C2): This interface is used to communicate with an SFF for various purposes (e.g., communicate required information for SFC forwarding decision-making, collect state information to adjust SFPs, collected connected SFs, etc.). Section 3.3.2 specifies such interface.

3. Interface between SFC Control Plane & SFC-aware SFs (C3): The SFC control plane uses this interface to interact with SFC-aware SFs. This interaction may be direct or via dedicated SF management systems (Section 3.3.3).
4. Interface between SFC Control Plane & SFC proxy (C4): The SFC control plane uses this interface to interact with an SFC proxy to communicate SFC instructions and to retrieve state information required, e.g., for dynamic SFP adjustments (Section 3.3.4).

This document does not elaborate on the internal decomposition of the SFC control plane functional blocks. The components within the SFC control plane and their interactions are out of scope.

Note, the SFC control plane must be able to invoke SFC OAM mechanisms, and to determine the results of OAM operations.

3.2. Centralized vs. Distributed

The SFC control plane can be (logically) centralized, distributed or a combination thereof. Whether one or multiple SFC Control Elements are enabled is deployment-specific. Nevertheless, the following comments can be made:

SFC management (including SFC monitoring and supervision): is likely to be centralized.

SFC mapping rules: i.e., service instructions to bind a flow to a service function chain and SFP are likely to be managed by a central SFC Control Element, but the resulting policies can be shared among several Control Elements. Note, these policies can be complemented with local information (e.g., an IPv4 address/IPv6 prefix assigned to a customer) because such information may not be available to the central entity but known only during network attachment phase.

Path computation: can be either distributed or centralized.

Distributed path computation means that the selection of the exact sequence of SFs that a packet needs to invoke (along with instances and/or SFF locator information) is a result of a distributed path selection algorithm executed by involved nodes. For some traffic engineering proposes, the SFP may be constrained by the control plane; as such, some SFPs can be fully specified (i.e., list all the SFF/SFs that need to be solicited) or partially specified (e.g., exclude some nodes, explicitly select which instance of a given SF needs to be invoked, etc.).

SFP resiliency (including restoration) refers to mechanisms to ensure high available service function chains. It includes means to detect node/link/path failures. Both centralized and distributed mechanism to ensure SFP resiliency can be envisaged.

Implementing a (logically) centralized path computation engine requires information to be dynamically communicated to the central SFC Control Element, such as the list of available SF instances, SFF locators, load status, SFP availability, etc.

3.3. Interface Reference Points

The following sub-sections describe the interfaces between the SFC control plane, as well as various SFC data plane elements.

3.3.1. C1: Interface between SFC Control Plane & SFC Classifier

As a reminder, a classifier is a function that is responsible for classifying traffic based on (pre-defined) rules.

This interface is used to install SFC classification rules in classifiers. Once classification rules are populated, classifiers are responsible for binding incoming traffic to service function chains and SFPs according to these classification rules. Note, the SFC control plane must not make any assumption on how the traffic is to be bound to a given service function chain. In other words, classification rules are deployment-specific. For instance, classification can rely on a subset of the information carried in a received packet such as 5-tuple classification, be subscriber-aware, be driven by traffic engineering considerations, or any combination thereof. Installing classification rules must be immediate. The status of enforcing such rules must be communicated to the control plane as part of the communication procedure. In particular, specific error codes must be returned to the Control Element in case an error is encountered during the enforcement procedure.

The SFC control plane should be responsible for removing invalid (and stale) mappings from the classification tables maintained by the classifiers. Also, local sanity checks mechanisms may be supported locally by the classifiers, but those are out of scope.

The classifier may be notified (regularly or upon eventual change) by the control plane about the available SFs (including the SFFs they are attached to) or be part of the service function discovery procedure.

Classification rules may be updated, deleted or disabled by the control plane. Criteria that would trigger those operations are deployment-specific.

This interface is also used to retrieve the list of classification rules that are maintained by a classifier. This retrieval can be on demand (at the initiative of the Control Element) or on a regular basis (at the initiative of the classifier).

Given that service function chaining solutions may be applied to very large sets of traffic, any control solution should take scaling issues into consideration as part of the design. For example, because a large number (e.g., 1000s) of classification entries may be configured to a classifier, means to reduce classification lookup time such as optimizing the size of the classification table (e.g., by means of aggregation capabilities) should be supported by the SFC control plane (and/or the classifier).

Below are listed some functional objectives that can be achieved thanks to the invocation of this interface:

- o Rationalize the management of classification rules.
- o Maintain a global view of instantiated rules in all classifiers in an SFC-enabled domain.
- o Check the consistency of instantiated classification rules within the same classifier or among multiple classifiers.
- o Assess the impact of removing or modifying a classification rule on packets entering an SFC-enabled domain.
- o Aggregate classification rules for the sake of performance optimization (mainly reduce lookup delays).
- o Adjust classification rules when rules are based on volatile identifiers (e.g., an IPv4 address, IPv6 prefix).
- o Allow to rapidly restore SFC/SFP states during failure events that occurred at a classifier (or a Control Element).

The control plane must instruct the classifier about the initial values of the Service Index (SI).

Also, the control plane must instruct the classifier about the set of metadata to be supplied in the context of a given chain.

SFC encapsulation protocol [I-D.ietf-sfc-nsh] includes metadata type 1 (MD#1) with mandatory context headers that can be used to convey metadata along an SFP. [I-D.ietf-sfc-nsh] allows defining different semantics in the context headers, but the NSH header does not convey that semantics in the context header. [I-D.ietf-sfc-nsh] requires an SFC-aware SF using the data placed in the MD#1 mandatory context headers to use information external to the NSH data plane to

understand the semantics of the context data. Therefore, this interface must provide such context semantics, including any suitable scoping information.

The control plane must instruct the classifier whether it can trust an existing SFC information carried in an incoming packet or whether it must be ignored.

A classifier should send unsolicited messages through this interface to notify the SFC control plane about specific events. Triggers for sending unsolicited messages should be configurable parameter.

When re-classification is allowed in an SFC-enabled domain, this interface can be used to control classifiers co-resident with SFC-aware SFs, SFC proxies, or SFFs to manage re-classification rules.

When an incoming packet matches more than one classification rule, tie-breaking criteria should be followed (e.g., priority). Such tie-breaking criteria should be instructed by the control plane.

The identification of instantiated SFCs/SFPs is local to each administrative domain; it is policy-based and deployment-specific.

3.3.2. C2: Interface between SFC Control Plane & SFF

SFFs make traffic forwarding decisions according to the entries maintained in their SFP Forwarding Policy Table. Such table is populated by the SFC control plane through the C2 interface. In particular, this interface is used to instruct the SFF about the set of information to use for lookup purposes (e.g., SFP-id, 5-tuple transport coordinates). One or many entries may be installed using one single control message. Installing new entries in the SFP Forwarding Policy Table must be immediate. The status of enforcing such entries must be communicated to the control plane as part of the communication procedure. In particular, specific error codes must be returned to the Control Element in case an error is encountered during the enforcement procedure.

This interface is used to instruct an SFF about the SFC-aware SFs that it can service. Such instruct typically occurs at the bootstrapping of the SFF, in the event of a new SF is added to the SFC-enabled domain, etc.

This interface is also used by the SFF to report the connectivity to their attached (including embedded) SFs. Local means may be enabled between the SFC-aware SFs and SFFs to allow for the dynamic attachment of SFs to an SFF and/or discovery of SFs by an SFF but those means are unspecified in this document.

The C2 interface is also used for collecting states of attributes (e.g., availability, workload, latency), for example, to dynamically adjust Service Function Paths. Such state can be collected using an explicit request from a Control Element or by unsolicited notification of the SFF on a regular basis or when an event occurs. A configuration parameter should be supported by the SFF to instruct the exact behavior to follow.

The C2 interface may be used to configure groups of functionally equivalent SFs. In particular, this group may be used for load-balancing purposes.

An SFF must be instructed to strip the SFC information for the chains it terminates. Forwarding policies for handling packets bound to chains that are terminated by an SFF may be communicated via this interface. By default, an SFF relies on legacy processing for forwarding these packets.

3.3.3. C3: Interface between SFC Control Plane & SFC-aware SFs

SFs may need to output some processing results of packets to the SFC control plane. This information can be used by the SFC control plane to update the SFC classification rules and the SFP Forwarding Policy Table entries.

This interface is used to collect such kind of feedback information from SFs. For example, the following information can be exchanged between an SF and the SFC control plane:

- o SF execution status: Some SFs may need to send information to the control plane to fine tune SFPs. For example, a threat-detecting SF can periodically send the threat characteristics via this interface, such as high probability of threat with packet of a given size. The control plane can then add an appropriate matching criteria to SFF to steer traffic to a scrubbing center.
- o SF load update: When SFs are under stress that yielded the crossing of some performance thresholds, the SFC control plane needs to be notified to adjust SFPs accordingly (especially when the centralized path computation mode is enabled). It is out of scope of this document to specify the exact methods to monitor the performance threshold or stress level of SFs, nevertheless the SFC control plane can invoke those methods for its operations.
- o SF bypass: An SF may use this interface to notify the Control Plane about its desire to be bypassed. The exact details about SF bypass logic are out of scope of this document.

The SFC control needs the above status information for various tasks it undertakes, but this information may be acquired directly from SFs or indirectly from other management and control systems in the operational environment.

This interface is used by an SFC-aware SF to report the set of context information (a.k.a., metadata) that it supports and any change of its capabilities, for example, as a result of a software update. Such change notifications should be dynamic, by default. A configuration parameter may be supported to disable such behavior.

This interface is also used to instruct an SFC-aware SF about any metadata it needs to attach to packets for a given SFC. This instruction may occur any time during the validity lifetime of an SFC/SFP.

Also, this interface informs the SFC-aware SF about the semantics of a context information, which would otherwise have opaque meaning. Several attributes may be associated with a context information such as (but not limited to) the "scope" (e.g., per-packet, per-flow, or per host), whether it is "mandatory" or "optional" to process flows bound to a given chain, etc. Note that a context may be mandatory for "chain 1", but optional for "chain 2". In particular, this interface must provide NSH MD#1 mandatory context semantics, including any suitable scoping information.

The control plane may indicate, for a given service function chain, an order for consuming a set of contexts supplied in a packet. This order may be indicated any time during the validity lifetime of an SFC/SFP.

An SFC-aware SF can also be instructed about the behavior it should adopt after consuming a context information that was supplied in the SFC header. For example, the context can be maintained, updated, or stripped.

Multiple SFs may be located within the same physical node, but no SFF is enabled in that same node, means to unambiguously forward the traffic from the SFF to the appropriate SF must be supported. Concretely, each SF must have a unique locator for unambiguous forwarding. This locator may be configured using this interface.

The controller may use the C3 interface to specify how the reverse path of flows, that are processed for a given direction, is selected by the SF. This feature is useful, for example, for packets generated by an SFC-aware SF to ensure these packets are forwarded to the corresponding source node with the same set of SFs, involved in the forward path, are invoked in the reverse order when forwarding

back these packets. Special care should be considered to avoid that instructions provided to distinct SFs lead to loops. Additional considerations are discussed in Section 4.2.

3.3.4. C4: Interface between SFC Control Plane & SFC Proxy

This interface is used by an SFC proxy to report the set of context information (a.k.a., metadata) that it supports and any change of its capabilities that may result, for example, in a software update. Such change notifications should be dynamic, by default. A configuration parameter may be supported to disable such behavior.

The SFC proxy can be instructed about authorized SFC-unaware SFs it can service. This instruction may occur during the bootstrapping of the SFC proxy or anytime during the SFP proxy operation time.

An SFC proxy may be instructed about the behavior it should adopt to process the context information that was supplied in the SFC header on behalf of an SFC-unaware SF, e.g., the context can be maintained or stripped.

The SFC proxy is also instructed about the semantics of a context information (including MD#1), which would otherwise have opaque meaning. Several attributes may be associated with a context information such as (but not limited to) the "scope" (e.g., per-packet, per-flow or per host), whether it is "mandatory" or "optional" to process flows bound to a given chain, etc.

The SFC proxy may also be instructed to add some new context information into the SFC header on behalf of an SFC-unaware SF.

The C4 interface is also used for collecting attribute states (e.g., availability, workload, latency), for example, to dynamically adjust Service Function Paths.

This interface may also be used to instruct the SFC proxy about the state and information to maintain for proper handling of packets received back from an SFC-unaware SF.

4. Additional Considerations

4.1. Discovery of the SFC Control Element

SFC data plane functional elements need to be provisioned with the locators of the Control Elements. This can be achieved using a variety of mechanisms such as static configuration or the activation of a service discovery mechanism. The exact specification of how this provisioning is achieved is out of scope.

4.2. SF Symmetry

Some SFs require both directions of a flow to traverse. Some service function chains require full symmetry. If an SF (e.g., stateful firewall or NAT) needs both direction of a flow, it is the SF instantiation that needs both direction of a flow to traverse, not the abstract SF (which can have many instantiations spread across the network).

Typically:

- o C1 interface is used to instruct the classifier how both directions of a flow should be processed when crossing an SFC-enabled domain.
- o C2 interface may be used to ensure that the same SF instance is involved in both directions of a flow (including, to ensure full chain symmetry).

4.3. Pre-deploying SFCs

Enabling service function chains should preserve some deployment practices adopted by Operators. Particularly, installing a service function chain (and its associated SFPs) should allow for pre-deployment testing and validation purposes (that is a restricted and controlled usage of such service function chain (and associated SFPs)).

4.4. Withdraw a Service Function (SF)

During the lifetime of an SFC, a given SF can be decommissioned. To accommodate such context and any other case where an SF is to be withdrawn, the control plane should instruct the SFC data plane functional element about the behavior to adopt. For example:

1. a first approach would be to update the service function chains and/or associated SFPs where that SF is present by removing any reference to that SF. The update concerns service function chains if the decommissioned SF is not provided by any active node. SFPs are impacted when alternate SF instances can provide the same service of the decommissioned SF instance.
2. a second approach would be to delete/deactivate any service function chain (and its associated SFPs) that involves that SF but install new service function chains.

4.5. SFC/SFP Operations

Various actions can be executed on a service function chain (and associated SFPs) that is structured by the SFC control plane. Indeed, a service function chain (and associated SFPs) can be enabled, disabled, its structure modified by adding a new SF hop or remove an SF from the sequence of SFs to be invoked, its classification rules modified, etc.

A modification of a service function chain can trigger control messages with the appropriate SFC-aware nodes accordingly.

The approach to be followed to migrate traffic to a new SFP from an old SFP is deployment-specific. For example, in order to avoid service disruption, a make-before-break mechanism can be followed where a new SFP is allocated to replace an existing SFP. Once the new SFP is set up, tested and the traffic is migrated to it, the old SFP can be removed. Other strategies may be followed within an SFC-enabled domain.

4.6. Unsolicited (Notification) Messages

SFC data plane functional elements must be instructed to send unsolicited notifications when loops are detected, a problem in the structure of a service function chain is encountered, a long unavailable forwarding path time is observed, etc.

Specific criteria to send unsolicited notifications to a Control Element should be fine tuned by the control plane using the interface defined in Section 3.3.

4.7. Liveness Detection

The control plane must allow to detect the liveness of SFC data plane elements of an SFC-enabled domain. Note that a data element may responsive from a connectivity standpoint, but the service it is supposed to provide may not be available.

In particular, the control plane must allow to dynamically detect that an SF instance is out of service and notify the relevant Control Element accordingly. The liveness information may be acquired directly from SFs or indirectly from other management and control systems in the operational environment.

Liveness status records for all SF instances, and service function chains (including the SFPs bound to a given chain) are maintained by the SFC Control.

The classifier may be notified by the control plane or be part of the liveness detection procedure.

The ability of an SFC Control Element to check the liveness of each SF present in service function chain has several advantages, including:

- o Enhanced status reporting by the control plane (i.e., an operational status for any given service chain derived from liveness state of its SFs).
- o Ability to support various resiliency policies (i.e., bypass a node embedding an SF, use alternate node, use alternate chain, drop traffic, etc.) .
- o Ability to support load balancing capabilities to solicit multiple SF instances that provide equivalent functions.

Local failure detect and repair mechanisms may be enabled by SFC-aware nodes. Control Elements may be fed directly or indirectly with inputs from these mechanisms.

Because a node embedding an SF can be responsive from a reachability standpoint (e.g., IP level) while the function it provides may be broken (e.g., a NAT module may be down), additional means to assess whether an SF is up and running are required. These means may be service-specific.

4.8. Monitoring & Counters

SFC-specific counters and statistics must be provided using the interfaces defined in Section 3.3. These data include (but not limited to):

- o Number of flows ever and currently assigned to a given service function chain and a given SFP.
- o Number of flows, packets, bytes dropped due to policy.
- o Number of packets and bytes in/out per service function chain and SFP.
- o Number of flows, packets, bytes dropped due to unknown service function chain (this is valid in particular for an SF node).

Even if setting the data collection cycle is deployment-specific, it is recommended to support dynamic means for better SFC automation.

4.9. Validity Lifetime

SFC instructions communicated via the various interfaces introduced in Section 3.3 may be associated with validity lifetimes, in which case classification and SFP Forwarding Policy Table entries will be

automatically removed upon the expiry of the validity lifetime without requiring an explicit action from a Control Element.

Lifetimes are used in particular by an SFC data plane element to clear invalid control entries that would be maintained in the system if, for some reason, no appropriate action was undertaken by the control plane to clear such entries.

Both short and long lifetimes may be assigned.

4.10. Considerations Specific to the Centralized Path Computation Model

This section focuses on issues that are specific to the centralized deployment model (Section 3.2).

4.10.1. Service Function Path Adjustment

An SFP is determined by composing SF instances and overlay links among SFFs. Thus, the status of an SFP depends on the states or attributes (e.g., availability, topological location, latency, workload, etc.) of its components. For example, failure of a single SF instance results in failure of the whole SFP. Since these states or attributes of SFP components may vary in time, their changes should be monitored and SFPs should be dynamically adjusted.

Examples of use cases for SFP adjustment are listed below:

SFP fail-over: re-construct an SFP with replacing the failed SF instance with another instance of the same SF or withdraw the failed SF from being invoked. Note that withdrawing an SF may be envisaged if the resulting connectivity service is not broken (that is, packets bound to the updated SFP can be successfully delivered to their ultimate destinations). Rerouting the traffic to another SF instance or withdrawing the failed SF is deployment-specific.

SFP with better latency experience: re-construct an SFP with a low path stretch considering the changes in topological locations of SF instances and the latency induced by the (overlay) connectivity among SFFs.

Traffic engineered SFP: re-construct SFPs to localize the traffic in the network considering various TE goals such as bypass a node, bypass a link, etc. These techniques may be used for planned maintenance operations on an SFC-enabled domain.

SF/SFP Load-balancing: re-construct SFPs to distribute the workload among various SF instances. Particularly, load distribution

policies can be taken into account by the Control Element to re-compute an SFP or be provisioned as attributes to SFPs that will be installed using the control interfaces (C2 interface, typically).

For more details about the use cases, refer to [I-D.lee-nfvrg-resource-management-service-chain].

The procedures for SFP adjustment may be handled by the SFC control plane as follows:

- o Collect and monitor states and attributes of SF instances and overlay links via the C2 interface (Section 3.3.2) and the C3 interface (Section 3.3.3).
- o Evaluate SF instances and overlay links based on the monitoring results.
- o Select SF instances to re-determine an SFP according to the evaluation results.
- o Replace target SF instances (e.g., in a failure or overladed) with newly selected ones.
- o Enforce the updated SFP for upcoming SFC traversal to SFFs via the C1 interface (Section 3.3.1) or the C2 interface (Section 3.3.2).

4.10.2. Head End Initiated SFP Establishment

In some scenarios where an SFC Control Element is not connected to all SFFs in an SFC-enabled domain, the SFC control plane can send the explicit SFF/SF sequence or SF sequence to the SFC head-end, e.g., the classifier via the C1 interface (Section 3.3.1). SFC head-end can use a signaling protocol to establish the SFF/SF sequence based on the SF sequence. Additional information (e.g., SF/SFF load) may be communicated to the SFC head-end to adjust an SFP.

4.10.3. (Regional) Restoration of Service Functions

There are situations that it might not be feasible for the classifier to be notified of the changes of SFF-sequence or SFF/SF Sequence for a given SFP because of the time taken for the notification and the limited capability of the classifiers.

If an SF has a large number of instantiations, it scales better if the classifier doesn't need to be notified with status of visible instantiations of SFs on an SFP.

It might not be always feasible for the classifier to be aware of the exact SF instances selected for a given SFP due to too many instances for each SF, notifications not being promptly sent to the classifier, or other reasons. This is about multiple instances of the same SF attached to one SFF node; those instances can be handled by the SFF via local load balancing schemes.

Regional restoration can take the similar approach as the global restoration: choosing a regional ingress node that can take over the responsibility of installing the new steering policies to the involved SFFs or network nodes. Typically, the regional ingress node should be:

- o on the data path of the flow of the given SFC;
- o in front of the relevant SFFs or network nodes that are impacted by the change of the SFP;
- o capable of encoding the detailed SFP to the Service Chain Header of data packets of the identified flow; and
- o capable of removing the detailed SFP encoding in data packets after all the impacted SFFs and network nodes completed the policy installation.

4.10.4. Fully Controlled SFF/SF Sequence for a SFP

This section discusses some information that can be exchanged over C2 interface (Section 3.3.2) when the SFC Control Element explicitly passes the steering policies to all SFFs for the SFF/SF sequence of a given SFC. In this model, each SFF doesn't need to signal other SFFs for the SFP.

The SFF nodes are not required to be directly adjacent to each other. As such, they can be interconnected using an overlay technique, such as Generic Routing Encapsulation (GRE), Virtual eXtensible Local Area Network (VXLAN), etc. SFs are attached to an SFF node or SFC proxy node via Ethernet link or other link types. As a local decision, there may be multiple different steering policies that work in conjunction with the SFC encapsulation [I-D.ietf-sfc-nsh] for one flow within one SFF.

For example, the semantics of traffic steering rules can be a match condition and an action, similar to, e.g., the route described in Section 2.3 of [I-D.ietf-i2rs-rib-info-model]. The match conditions and action for distinct ports can be different.

The matching criteria for SFF can be more sophisticated. For example, it could be the SFP-id carried within the SFC encapsulation with any fields in the data packets, such as (non-exhaustive list):

- o Destination MAC address
- o Source MAC address
- o VLAN-ID,
- o Destination IP address
- o Source IP address
- o Source port number
- o Destination port number
- o Differentiated Services Code Point (DSCP)
- o Packet size, etc., or any combination thereof.

An SFF node may not support some of the matching criteria listed above. It is important that SFC control plane can retrieve the supported matching criteria by SFF nodes. The actions for traffic steering could be to steer traffic to the attached SF instances via a specific port.

The actions to SFC proxy may include a method to map the SFP identifier carried in the packet header to a locally significant link identifier, e.g., VLAN-ID, and a method to construct and encapsulate the SFC header back to the packets when they come back from the attached SFs.

This approach does not require using an end-to-end signaling protocol among classifier nodes and SFF nodes. However, there may be problems encountered if SFF nodes are not updated in the proper order or not at the same time. For example, if the SFF "A" and SFF "C" get flow steering policies at slightly different times, some packets might not be directed to some service functions on a chain.

5. Security Considerations

5.1. Secure Communications

The SFC Control Elements and the participating SFC data plane elements must mutually authenticate. SFC data plane elements must ignore instructions received from unauthenticated SFC Control Elements. The credentials details used during authentication can be used by the SFC control plane to decide whether specific authorization may be granted to a Service Function with regards to some specific operations (e.g., authorize a given SF to access specific context information).

In case multiple SFC data plane elements are embedded in the same node, the authentication mechanism may be executed as a whole; not for each instance.

An SFC data plane element must be able to send authenticated unsolicited notifications to an SFC Control Element.

The communication between a Control Element and SFC data plane elements must provide integrity and replay protection.

A Service Function must by default discard any action from an SFC Control Element that requires specific right privileges (e.g., access to a legal intercept log, mirror the traffic, etc.).

5.2. Pervasive Monitoring

The authentication mechanism should be immune to pervasive monitoring [RFC7258]. An attacker can intercept traffic by installing classification rules that would lead to redirect all or part of the traffic to an illegitimate network node. Means to protect against attacks that would lead to install, remove, or modify classification rules must be supported.

5.3. Privacy

The SFC control plane must be able to instruct SFC data plane elements about the information to be leaked outside an SFC-enabled domain. Particularly, the SFC control plane must support means to preserve privacy [RFC6973]. Context headers may indeed reveal privacy information (e.g., IMSI, user name, user profile, location, etc.). Those headers must not be exposed outside the operator's domain.

5.4. Denial-of-Service (DoS)

In order to protect against denial of service that would be caused by a misbehaving trusted SFC Control Element, SFC data plane elements should rate limit the messages received from an SFC Control Element.

5.5. Illegitimate Discovery of SFs and SFC Control Elements

Means to defend against soliciting illegitimate SFs/SFFs that do not belong to the SFC-enabled domain must be enabled. Such means must be defined in service function discovery and SFC Control Element discovery specification documents.

6. IANA Considerations

This document does not require any IANA actions.

7. Acknowledgments

This document is the result of merging with [I-D.lee-sfc-dynamic-instantiation].

Hongyu Li, Qin Wu, and Yong(Oliver) Huang edited an early version of the individual submission of this document.

Many thanks to Shibi Huang, Lac Chidung, Taeho Kang, Sumandra Majee, Dave Dolson, Paul Bottorff, Reinaldo Penno, Jim Guichard, Shunsuke Homma, Ken Gray, Henry Fourie, and Dirk von Hugo for the feedback and discussion on the mailing list.

The text about the semantic of a context information is provided by Dave Dolson and Lucy Yong.

Many thanks to Paul Quinn and Uri Elzur for the detailed review.

Thanks to Catherine Meadows for the SecDir review, and to Stephen Farrell and Tero Kivinen for scheduling an early SecDir review.

Special thanks to Alia Atlas for the careful AD review.

8. Contributors

The following individuals have contributed significantly to this document:

Hongyu Li
Huawei
Huawei Industrial Base, Bantian, Longgang
Shenzhen
China

EMail: hongyu.li@huawei.com

Qin Wu
Huawei
101 Software Avenue, Yuhua District
Nanjing, Jiangsu 210012
China

EMail: bill.wu@huawei.com

Yong(Oliver) Huang
Huawei
Huawei Industrial Base, Bantian, Longgang
Shenzhen
China

EMail: oliver.huang@huawei.com

Christian Jacquenet
Orange
Rennes 35000
France

EMail: christian.jacquenet@orange.com

Walter Haeffner
Vodafone D2 GmbH
Ferdinand-Braun-Platz 1
Duesseldorf 40549
DE

EMail: walter.haeffner@vodafone.com

Seungik Lee
ETRI
218 Gajeong-ro Yuseung-Gu
Daejeon 305-700
Korea

Phone: +82 42 860 1483
EMail: seungiklee@etri.re.kr

Ron Parker
Affirmed Networks
Acton
MA 01720
USA

EMail: ron_parker@affirmednetworks.com

Linda Dunbar
Huawei Technologies
USA

EMail: ldunbar@huawei.com

Andrew Malis
Huawei Technologies
USA

EMail: agmalis@gmail.com

Joel M. Halpern
Ericsson

EMail: joel.halpern@ericsson.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Sarjapur Marathalli Outer Ring Road
Bangalore, Karnataka 560103
India

EMail: tireddy@cisco.com

Prashanth Patil
Cisco Systems, Inc.
Bangalore
India

EMail: praspati@cisco.com

9. References

9.1. Normative References

[RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

9.2. Informative References

[I-D.ietf-i2rs-rib-info-model]
Bahadur, N., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-09 (work in progress), July 2016.

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

- [I-D.ietf-sfc-dc-use-cases]
Surendra, S., Tufail, M., Majee, S., Captari, C., and S. Homma, "Service Function Chaining Use Cases In Data Centers", draft-ietf-sfc-dc-use-cases-05 (work in progress), August 2016.
- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [I-D.ietf-sfc-use-case-mobility]
Haeffner, W., Napper, J., Stiemerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", draft-ietf-sfc-use-case-mobility-07 (work in progress), October 2016.
- [I-D.lee-nfvrg-resource-management-service-chain]
Lee, S., Pack, S., Shin, M., and E. Paik, "Resource Management in Service Chaining", draft-lee-nfvrg-resource-management-service-chain-01 (work in progress), March 2015.
- [I-D.lee-sfc-dynamic-instantiation]
Lee, S., Pack, S., Shin, M., and E. Paik, "SFC dynamic instantiation", draft-lee-sfc-dynamic-instantiation-01 (work in progress), October 2014.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<http://www.rfc-editor.org/info/rfc3022>>.
- [RFC3135] Border, J., Kojo, M., Griner, J., Montenegro, G., and Z. Shelby, "Performance Enhancing Proxies Intended to Mitigate Link-Related Degradations", RFC 3135, DOI 10.17487/RFC3135, June 2001, <<http://www.rfc-editor.org/info/rfc3135>>.
- [RFC6146] Bagnulo, M., Matthews, P., and I. van Beijnum, "Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers", RFC 6146, DOI 10.17487/RFC6146, April 2011, <<http://www.rfc-editor.org/info/rfc6146>>.
- [RFC6333] Durand, A., Droms, R., Woodyatt, J., and Y. Lee, "Dual-Stack Lite Broadband Deployments Following IPv4 Exhaustion", RFC 6333, DOI 10.17487/RFC6333, August 2011, <<http://www.rfc-editor.org/info/rfc6333>>.

- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<http://www.rfc-editor.org/info/rfc6973>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.

Author's Address

Mohamed Boucadair (editor)
Orange
Rennes
35000
France

EMail: mohamed.boucadair@orange.com

sfc
Internet-Draft
Intended status: Standards Track
Expires: May 4, 2017

R. Maglione
G. Trueba
C. Pignataro
Cisco Systems
October 31, 2016

RADIUS Attributes for NSH
draft-maglione-sfc-nsh-radius-01

Abstract

Network Service Header (NSH) protocol defines the Service Function Chaining (SFC) encapsulation required to support the Service Function Chaining (SFC) Architecture. One of the components of the Network Service Header (NSH) protocol is the Service Path Identifier (SPI), which identifies a service path, another important element of the NSH protocol is the Service Index (SI), which provides location within the Service Path.

When Service Providers would like to deliver customized services offers requiring Service Functions Chains, a different service chain may be required for each subscriber or group of subscribers. In order to simplify the service provisioning in this scenario, it would be useful to be able to associate the Service Path Identifier (SPI), identifying the service chain, and the appropriate Service Index (SI), identifying the location in the service path, with the customer profile.

In some Broadband networks, the customer profile information may be stored in Authentication, Authorization, and Accounting (AAA) servers. This document specifies two new Remote Authentication Dial-In User Service (RADIUS) attributes to carry the Service Path Identifier (SPI) and the Service Index (SI).

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 4, 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	3
2. Terminology	3
3. Architectural Model	3
4. RADIUS Attributes	4
4.1. NSH Service Path Identifier	5
4.2. NSH Service Index	6
5. Table of Attributes	7
6. Diameter Considerations	8
7. Acknowledgements	8
8. IANA Considerations	8
9. Security Considerations	8
10. References	9
10.1. Informative References	9
10.2. Normative References	9
Authors' Addresses	10

1. Introduction

Network Service Header (NSH) protocol [I-D.ietf-sfc-nsh] defines the Service Function Chaining (SFC) encapsulation required to support the Service Function Chaining (SFC) Architecture [RFC7665]. One of the components of the Network Service Header (NSH) protocol is the Service Path Identifier (SPI), which identifies a service path, another important element of the NSH protocol is the Service Index (SI), which provides location within the Service Path.

When Service Providers would like to deliver customized services offers requiring Service Functions Chains, a different service chain may be required for each subscriber or group of subscribers. In order to simplify the service provisioning in this scenario, it would be useful to be able to associate the Service Path Identifier (SPI), identifying the service chain, and the appropriate Service Index (SI) identifying the location in the service path, with the customer profile.

In some Broadband networks, the customer profile information may be stored in Authentication, Authorization, and Accounting (AAA) servers. This document specifies two new Remote Authentication Dial-In User Service (RADIUS) attributes to carry the Service Path Identifier (SPI) and the Service Index (SI).

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].

2. Terminology

NSH Network Service Header
SFC Service Function Chaining
SFF Service Function Forwarder
SPI Service Path Identifier
SI Service Index

3. Architectural Model

Figure 1 illustrates the network reference model for a Broadband access scenario where a NAS, acting as RADIUS Client, performs both the Service Classification and Service Forwarder Function.

The Service Functions which make up the Service Chaining are part of the SP network and they are not depicted in Figure 1

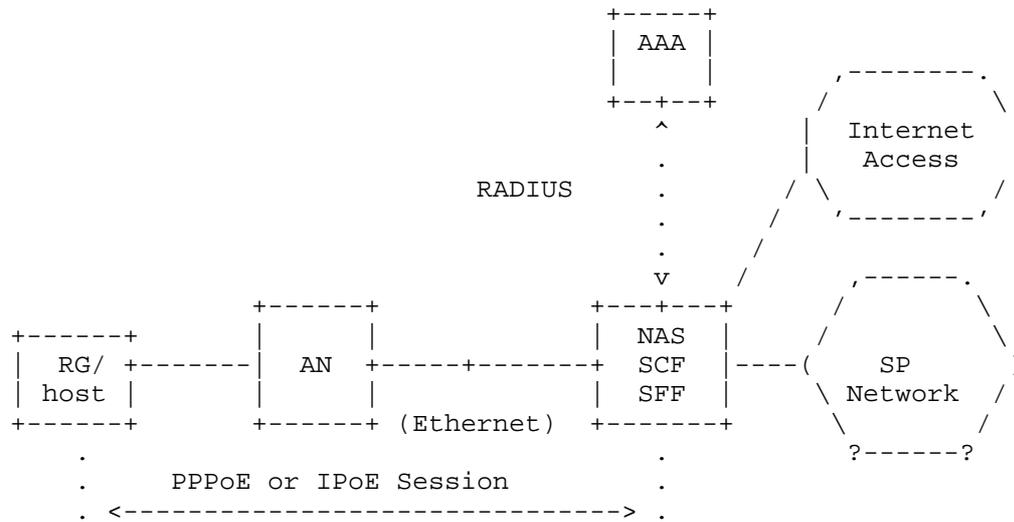


Figure 1: Network Reference Model

Here there is a brief description of the Authentication/Authorization process between the NAS and the AAA Server.

The NAS initially sends a RADIUS Access-Request message to the RADIUS server, requesting authentication. Once the RADIUS server receives the request, it validates the sending client, and if the request is approved, the AAA server replies with an Access-Accept message including a list of attribute-value pairs that describe the parameters to be used for this session. This list MAY also contain the NSH Service Path Identifier (NSH-SPI) and the NSH Service Index (NSH-SI) attributes used to identify a specific service path and the location in the service path.

The NSH SPI attribute returned by AAA Server in the Access-Accept is used by the NAS to insert the traffic of the subscriber in the correct service path. A classification rule, to be associated with the SPI, can also be sent by the AAA Server as part of the list of attribute-value pairs.

4. RADIUS Attributes

This section defines the NSH Service Path Identifier (SPI) and the NSH Service Index (SI) attributes that are used in the above-mentioned scenario. The attributes design follows [RFC6158] and refers to [RFC6929] and [I-D.ietf-radext-datatypes].

4.1. NSH Service Path Identifier

The NSH Service Path Identifier (NSH-SPI) RADIUS attribute contains the value which identifies a specific service path to be associated to a subscriber.

When the NAS receives from the AAA Server the NSH-SPI attribute, the NAS MUST use the value contained in this attribute to populate the Service Path Identifier (SPI) field in the NSH Service Path header defined in [I-D.ietf-sfc-nsh].

If the NAS is pre-configured with a default NSH SPI value, this value MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS, and it MAY assign a different NSH SPI.

If the NAS includes the NSH-SPI attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA server. If the NAS does not receive the NSH-SPI attribute in the Access-Accept message, it MAY fall back to a pre-configured default NSH SPI, if any. If the NAS receives the NSH-SI attribute, but it does not receive the NSH-SPI attribute from the AAA Server and the NAS does not have any pre-configured SPI, the traffic generated by that specific subscriber MUST be dropped as this is an error condition. If the NAS does not receive the NSH-SPI attribute and it does not receive the NSH-SI attribute in the Access-Accept message and the NAS does not have any pre-configured NSH SPI and NSH SI, the traffic generated by that specific subscriber does not have to be sent across any service chain.

If the NAS is pre-provisioned with a default NSH SPI and the NSH-SPI received in the Access-Accept message is different from the configured default, then the NSH-SPI received in the Access-Accept message MUST be used for the session.

If an implementation includes Change-of-Authorization (CoA) messages [RFC5176], they could be used to modify the current specified SPI. When the NAS receives a CoA Request message containing the NSH-SPI attribute, the NAS MUST use the received NSH SPI value to re-configure the the Service Path Identifier (SPI) field in the NSH Service Path header. This allows the network administrator to modify the forwarding of the traffic of a specific subscriber. By changing the SPI value the service path used for the subscriber is modified, thus the traffic of the selected subscriber is sent across a different service chain.

The NSH-SPI RADIUS attribute MUST NOT appear more than once in a message.

A summary of the NSH-SPI RADIUS attribute format is shown below. The fields are transmitted from left to right.

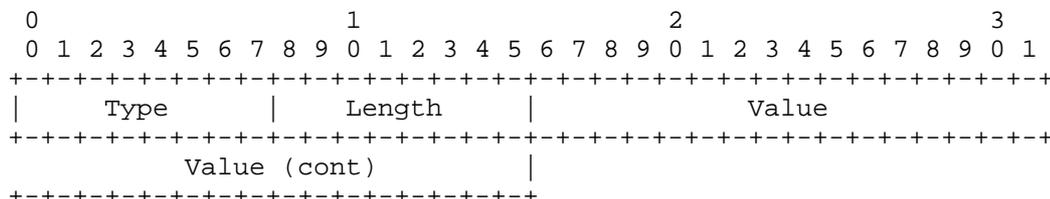


Figure 2: NSH-SPI RADIUS Attribute

Type TBD - NSH-SPI

Length 6

Value This field uses the integer data type, defined in [I-D.ietf-radext-datatypes], which encodes a 32-bit unsigned integer in network byte order. As the Service Path Identifier field, defined in [I-D.ietf-sfc-nsh], is limited to 24 bits, only 24 bits of the value field in the RADIUS attribute are used to encode the NSH-SPI value. The NAS acting as classifier MUST copy this value into the SPI field of the NSH Service Path Header.

4.2. NSH Service Index

The NSH Service Index (NSH-SI) RADIUS attribute contains the value which identifies the location in the service path. According to [I-D.ietf-sfc-nsh], the initial SI value SHOULD default to 255.

When the NAS receives from the AAA Server the NSH-SI attribute, the NAS MUST use the value contained in this attribute to populate the Service Index (SI) field in the NSH Service Path header defined in [I-D.ietf-sfc-nsh].

If the NAS is pre-configured with a default NSH SI value, this value MAY be inserted in the attribute. The RADIUS server MAY ignore the hint sent by the NAS, and it MAY assign a different NSH SI.

If the NAS includes the NSH-SI attribute, but the AAA server does not recognize it, this attribute MUST be ignored by the AAA server. If the NAS does not receive the NSH-SI attribute in the Access-Accept message, but it receives the NSH-SPI attribute, it MAY fall back to a pre-configured default NSH SI, if any. If the NAS receives the NSH-SPI attribute, but it does not receive the NSH-SI attribute from the AAA Server and the NAS does not have any pre-configured SI, the

traffic generated by that specific subscriber MUST be dropped as this is an error condition.

If the NAS is pre-provisioned with a default NSH SI and the NSH-SI received in the Access-Accept message is different from the configured default, then the NSH-SI received in the Access-Accept message MUST be used for the session.

If an implementation includes Change-of-Authorization (CoA) messages [RFC5176], they could be used to modify the current specified NSH SI. When the NAS receives a CoA Request message containing the NSH-SI attribute, the NAS MUST use the received NSH SI value to re-configure the the Service Index (SI) field in the NSH Service Path header.

The NSH-SI RADIUS attribute MUST NOT appear more than once in a message.

A summary of the NSH-SI RADIUS attribute format is shown below. The fields are transmitted from left to right.

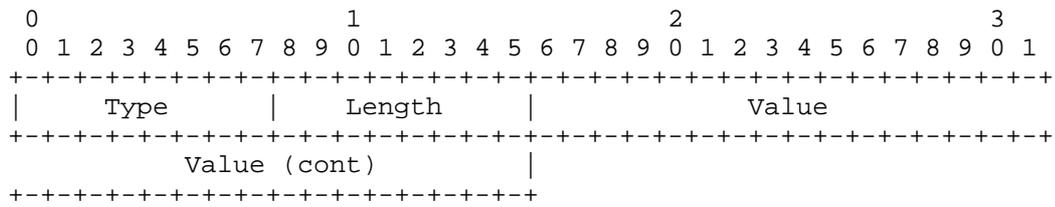


Figure 3: NSH-SI RADIUS Attribute

Type TBD - NSH-SI

Length 6

Value This field uses the integer data type defined in [I-D.ietf-radext-datatypes], which encodes a 32-bit unsigned integer in network byte order. As the Service Index field defined in [I-D.ietf-sfc-nsh] is limited to 8 bits, only 8 bits of the value field in the RADIUS attribute are used to encode the NSH-SI value. The NAS acting as classifier MUST copy this value into the SI field of the NSH Service Path Header.

5. Table of Attributes

The following tables provide a guide to which attributes may be found in which kinds of packets, and in what quantity.

Access-Request	Access-Accept	Access-Reject	Challenge	Accounting # Request	Attribute
0-1	0-1	0	0	0-1	TBD NSH-SPI
0-1	0-1	0	0	0-1	TBD NSH-SI

CoA-Request	CoA-ACK	CoA-NACK	#	Attribute
0-1	0	0		TBD NSH-SPI
0-1	0	0		TBD NSH-SI

The following table defines the meaning of the above table entries.

0 This attribute MUST NOT be present in the packet.

0+ Zero or more instances of this attribute MAY be present in the packet.

0-1 0-1 Zero or one instance of this attribute MAY be present in the packet.

6. Diameter Considerations

These attributes are usable within either RADIUS or Diameter [RFC6733]. Since the attributes defined in this document have been allocated from the standard RADIUS type space, no special handling is required by Diameter entities.

7. Acknowledgements

The authors would like to thank Jim Guichard and Mohamed Boucadair for their valuable comments and inputs to this document.

8. IANA Considerations

Per this document, IANA is requested to assign two new RADIUS Attribute Type in the "Radius Types" registry (currently located at <http://www.iana.org/assignments/radius-types>) for the following attributes:

TBD NSH-SPI integer

TBD NSH-SI integer

9. Security Considerations

This document has no additional security considerations beyond those already identified in [RFC2865] for the RADIUS protocol and in [RFC5176] for CoA messages.

The security considerations for NSH protocol are described in section 9 of [I-D.ietf-sfc-nsh]

10. References

10.1. Informative References

- [RFC5176] Chiba, M., Dommety, G., Eklund, M., Mitton, D., and B. Aboba, "Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)", RFC 5176, DOI 10.17487/RFC5176, January 2008, <<http://www.rfc-editor.org/info/rfc5176>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

10.2. Normative References

- [I-D.ietf-radext-datatypes] DeKok, A., "Data Types in the Remote Authentication Dial-In User Service Protocol (RADIUS)", draft-ietf-radext-datatypes-08 (work in progress), October 2016.
- [I-D.ietf-sfc-nsh] Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, DOI 10.17487/RFC2865, June 2000, <<http://www.rfc-editor.org/info/rfc2865>>.
- [RFC6158] DeKok, A., Ed. and G. Weber, "RADIUS Design Guidelines", BCP 158, RFC 6158, DOI 10.17487/RFC6158, March 2011, <<http://www.rfc-editor.org/info/rfc6158>>.
- [RFC6733] Fajardo, V., Ed., Arkko, J., Loughney, J., and G. Zorn, Ed., "Diameter Base Protocol", RFC 6733, DOI 10.17487/RFC6733, October 2012, <<http://www.rfc-editor.org/info/rfc6733>>.

[RFC6929] DeKok, A. and A. Lior, "Remote Authentication Dial In User Service (RADIUS) Protocol Extensions", RFC 6929, DOI 10.17487/RFC6929, April 2013, <<http://www.rfc-editor.org/info/rfc6929>>.

Authors' Addresses

Roberta Maglione
Cisco Systems
Via Torri Bianche 8
Vimercate
Italy

Email: robmgl@cisco.com

Guillermo Trueba
Cisco Systems
Avenida Cortes Valencianas 58
Valencia 46015
Spain

Email: gtrueba@cisco.com

Carlos Pignataro
Cisco Systems
7200 Kit Creek Road
Research Triangle Park, NC 27709
USA

Email: cpignata@cisco.com

SFC Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2017

D. Migault, Ed.
Ericsson
C. Pignataro
T. Reddy
Cisco
C. Inacio
CERT/SEI/CMU
October 28, 2016

SFC environment Security requirements
draft-mglt-sfc-security-environment-req-02.txt

Abstract

This document provides environment security requirements for the SFC architecture. Environment security requirements are independent of the protocols used for SFC - such as NSH for example. As a result, the requirements provided in this document are intended to provide good security practices so SFC can be securely deployed and operated. These security requirements are designated as environment security requirements as opposed to the protocol security requirements.

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 1, 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. Requirements notation 2
- 2. Introduction 2
- 3. Terminology and Acronyms 3
- 4. SFC Environment Overview 3
 - 4.1. Deployment of SFC Architecture 6
- 5. Threat Analysis 7
 - 5.1. Attacks performed from the SFC Control Plane 8
 - 5.2. Attacks performed from the SFC Management Plane 9
 - 5.3. Attacks performed from the Tenant's Users Plane 9
 - 5.4. Attacks performed from the SFC Data Plane 11
- 6. Security Requirements 14
 - 6.1. Plane Isolation Requirements 15
 - 6.1.1. SFC Control Plane Isolation 16
 - 6.1.2. SFC Management Plane Isolation 17
 - 6.1.3. Tenant's Users Data Plane Isolation 18
 - 6.2. SFC Data Plane Requirements 19
 - 6.3. Additional Requirements 22
- 7. Security Considerations 22
- 8. Privacy Considerations 23
- 9. IANA Considerations 23
- 10. Acknowledgments 23
- 11. References 23
 - 11.1. Normative References 23
 - 11.2. Informative References 24
- Authors' Addresses 24

1. Requirements notation

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].

2. Introduction

This document provides environment security requirements for the SFC architecture [I-D.ietf-sfc-architecture]. Environment security requirements are independent of the protocols used for SFC - such as NSH [I-D.ietf-sfc-nsh]. As a result, the requirements provided in this document are intended to provide good security practice so SFC

can be securely deployed and operated. These security requirements are designated as environment security requirements as opposed to the protocol security requirements. This document is built as follows. Section 4 provides an overall description of the SFC environment with the introduction of the different planes (SFC Control Plane, the SFC Management Plane, the Tenant's user Plane and the SFC Data Plane). Section 6 lists environment security requirements for the SFC. These requirements are intended to prevent attacks, as well as network and SFC misconfigurations. When such events happens, the security recommendations also aim at detecting and identifying the threats or misconfiguration as well as limiting their impact. Recommendations also may apply differently depending on the infrastructure. For example trusted environment may enforce lighter security recommendations than public and open SFC infrastructures. However, one should also consider future evolution of their infrastructure, and consider the requirements as a way to maintain the SFC architecture stable during its complete life cycle. For each requirement this document attempts to provide further guidance on the reasons to enforce it as well as what should be considered while enforcing it or the associated risks of not enforcing it.

This document assumes the reader is familiar with the SFC architecture defined in [I-D.ietf-sfc-architecture] as well as the Internet Security Glossary [RFC4949]

3. Terminology and Acronyms

In addition to the terminology defined in [I-D.ietf-sfc-architecture], the document defines the following terminology:

- Tenant: A tenant is one organization that is using SFC. A tenant may use SFC on one's own private infrastructure or on a shared infrastructure.
- Tenant's User Data Plane: The tenant may be using SFC to provide service to its customers or users. The communication of these users is designated as Tenant's user Data Plane and includes all communications involving the tenant's users. As a result, if a user is communicating with a server or a user from another domain, the communication with that tenant's user is part of the Tenant's Users Data Plane.

4. SFC Environment Overview

This section provides an overview of SFC. It is not in the scope to this document to provide an explicit description of SFC. Instead, the reader is expected to read [RFC7498],

[I-D.ietf-sfc-architecture], [I-D.ietf-sfc-control-plane] and other SFC related documents.

Service Function Chaining (SFC) architecture is defined in [I-D.ietf-sfc-architecture]. This section briefly illustrates the main concepts of the SFC architecture and positions the architecture within an environment.

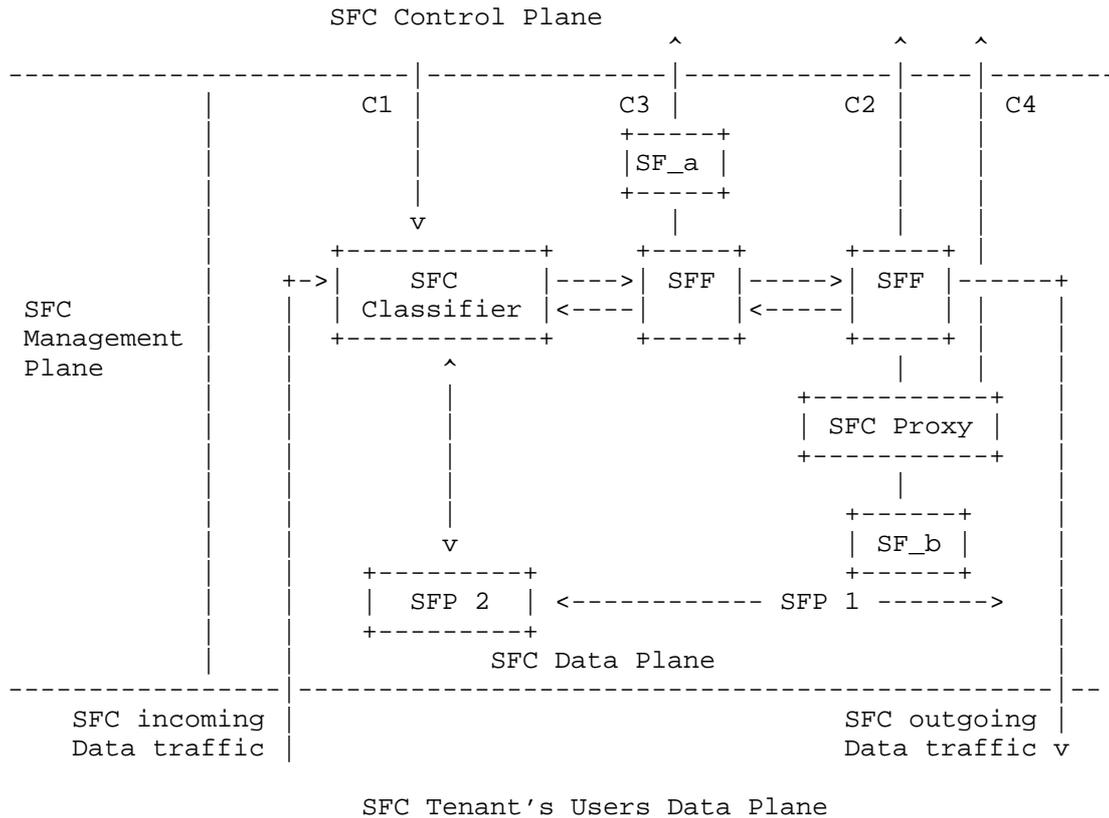


Figure 1: SFC Environment Overview

SFC defined a Service Function Path (SFP) which is an ordered set of Service Functions (SF) applied to part of the packets. The figure above represents two SFP: SFP1 and SFP2. SFP2 is not detailed but SFP1 defines a path that goes through SF_a and SF_b. SFP is defined at the SF level, which means the path does not consider the specific instance of an SF for example. A SF may be performed by different instances of SF located at different positions. As a result, a specific packet may pass through different instances of SFC. The

ordered set of SF instances a packet goes through is called the Rendered SF Path (RSFP).

Upon the receipt of an incoming packet from the tenant's user, the SFC Classifier determines, according to Classifiers, which SFP is associated to that packet. The packet is forwarded from Service Function Forwarders (SFF) to SFP. SFP are then in charge of forwarding the packet to the next SFF or to a SF. Forwarding decisions may be performed using SFP information provided by the SFC Encapsulation. As described in [I-D.ietf-sfc-nsh] the SFC Encapsulation contains SFP information such as the SFP ID and Service Index and eventually (especially for the MD-2 in NSH) some additional metadata. SF may be SFC aware or not. In the case the SFC functions are not SFC aware, a SFC Proxy performs the SFC Decapsulation (resp. SFC Encapsulation) before forwarding the packet to the SF (resp. after receiving the packet from the SF).

The environment associated to SFC may be separated into the four main planes:

- SFC Management Plane and Control Plane are defined in [I-D.ietf-sfc-control-plane]. The SFC Management Plane can be assimilated to the cloud infrastructure provider allocating various resource to the various SF and eventually active the various SF components. Typically management operations would consist in setting the number of CPU, memory bandwidth associated to the various SFs as well as specific configuration parameters of the SFC components. It is expected that the interface between the various SFC components configuration will be vendor specific. These configurations may be provided by the Cloud infrastructure provider or in the case of multitenancy by the administrator of the virtual network, or by each administrator of the SFC components. The SFC control plane controls and configure the SFC related components. The Control Plane differs from the Management Plane as it only concerns a subset of the parameters and facilities associated to the SF. In general, these parameters are expected to only modify the internal states of the different elements. This aspect confers programmability properties to the Control Plane that are usually not provide to the Management Plane. It is also expected that the SFP are elaborated in this plane before being pushed into the SFC Data Plane, and more generally, the SFP state in the SFF is expected to come from control rather than management.
- SFC Data Plane consists in all SF components as well as the data exchanged between the SF components. Communications between SF components includes the packet themselves, their

associated metadata, the routing logic - similar to RIB - or SF logic, i.e. what they returned values are for example. In other words, the SFC Data Plane can also be seen as all the elements that interact with a packet provided by an end user. Of course the end user is not expected to configure any of these element through the SFC Data Plane. Instead it is expected to apply the policies and configurations put in place by the SFC Tenant.

- SFC Tenant's Users Data Plane consists in the traffic data provided by the different users of the tenants. When a user is communicating with a server or another user -eventually from another administrative domain - , the communication belongs to the SFC Tenant's Users Data Plane whenever packets are provided by the server or by the user.

4.1. Deployment of SFC Architecture

This section illustrates a deployment of SFC we consider in this document.

A Cloud Provider provides an infrastructure that is shared by multiple SFC Tenants. The Cloud Provider may also provide some servers or hardware that have a dedicated function. Such hardware may be provided to the SFC Tenants under the form of a SF. It may thus be shared by multiple SFC Tenants. Such SF are designated as third party SF. Another case of SF may also consider a local SF proxying the traffic to a remote site or domain. The SF proxy transparently to the SFC elements may forward the traffic out of the boundaries of the Tenant. In some case this may be needed, but in some other case this may be done unbeknownst to the Tenant's.

Each SFC Tenant is responsible of its domain, that is to administrate or provision the necessary resource and control all its SFC elements which include defining SFC Paths, configuring the elements... Typically the coordination of the SFC elements is likely to be performed by a SDN controller.

Protecting the deployed SFC architecture from attacker is one goal of the security requirements. Some could easily argue that such requirements are not needed for example in a private SFC deployment where SFC components may be considered in a trusted environment and administrated by a single entity. However, even in a single administrative domain, inside attacks are possible. (e.g. inside attacker sniffing the SFC metadata, sending spoofed packets etc.). Then, the trusted domain assumption may not remain valid over time. Suppose, for example, that the SFC architecture is now interconnected with some third party SF or SFF. Such SFC component is now outside the initial trusted domain which has several security implications.

Similarly, a single trusted domain with one tenant may evolve over time and become multitenants and share a SFC platform. These tenants, may be trusted as in the case for example where each tenant represents a different department of a single company. Authentication is not sufficient, and relying only on a access control presents some risks. If the tenants are not strongly isolated - with physical or logical networks isolation, they may share a common SFP and one tenant may update the SFP of the other tenant. Such misconfiguration has similar impact as a redirecting attack. This document provide guidance that result in limiting such risks and improve detection for further mitigation.

5. Threat Analysis

The SFC environment is composed of the following plans: SFC Management Plane, SFC Control Plane, SFC Data Plane and SFC Tenant's User Data Plane. The purpose of these planes is to group a given set of functions while limiting the interactions between these planes. Interactions between planes are only limited - in most cases controlled - but these interactions still exist and so may be used by an attacker. As a result, for each plane, the threat analysis is performed by analysis the vulnerabilities present within each plane as well as those performed via the other planes.

Threat analysis of the Management Plane and the Control Plane have been described in [I-D.ietf-sfc-control-plane]. The SFC Tenant's User Plane is out of the boundaries of the SFC administrator. As a result attacks performed on SFC Tenant's User Plan are not considered in this section and this section limits its analysis on teh SFC Data Plan.

This section describes potential threats the SFC Data Plane may be exposed. The list of threats is not expected to be complete. More especially, the threats mentioned are provided to illustrate some security requirements for the SFC architecture. For simplicity, this document mostly considers that security breaches are performed by an attacker. However, such breaches may also be non-intentional and may result from misconfiguration for example.

Attacks may be performed from inside the SFC Data Plane or from outside the SFC Data plane, in which case, the attacker is in at least one of the following planes: SFC Control Plane, SFC Management Plane or SFC Tenants' Users Plane. Some most sophisticated attacks may involve a coordination of attackers in multiple planes.

5.1. Attacks performed from the SFC Control Plane

Attacks related to the control plane have been detailed in section 5 of [I-D.ietf-sfc-control-plane].

The different interfaces between the SFC Control Plane and the SFC Data Plane are exposed in [draft-ietf-sfc-control-plane]. It includes:

- Updating the classification rule of the SFC Classifier (also referred as interface C1).
- Updating the forwarding decision of the SFF (also referred as interface C2). This interface is also used to provide the SFC Control Plane some information for example on the system load, network load or the latency so appropriated SFP may be computed.
- Updating SF's internal states (interface C3). This interface is also used to provide the SFC Control Plane some information for example on the system load, network load or the latency so appropriated SFP may be computed.
- Updating SFC Proxy's internal states (interface C4). This interface is also used to provide the SFC Control Plane some information for example on the system load, network load or the latency so appropriated SFP may be computed.

An attacker may change the SFC Classifier classification and completely modify the services provided by the SFC. Such privileges may be used to avoid some control over the tenant's traffic (like firewalling service). An attacker may also modify the filtering or classification rules to overload heavy processing functions with traffic. In a pay-what-you-use model, this could result in extra cost for the tenant or to trigger a DoS attack on the tenant SFC Data Plane.

Attack performed on the SFC Control Plane mostly consists in tenant impersonation or communication hijacking. This would enable an attacker to control the SFC components associated to the tenant. Similarly an attacker may also collect system or network load information in order to better orchestrate a DoS attack for example. An attacker may also inject instructions in order to perform a DoS attack on a given SFC component or to prevent the tenant to control other SFC components.

5.2. Attacks performed from the SFC Management Plane

Attacks performed on the SFC Management Plane are similar to those performed from the SFC Control Plane. The main difference is that the SFC Management Plan provides usually a greater control of the SFC component than the SFC Control Plane.

In addition, the actions performed by the SFC Management Plane have fewer restrictions, which means it may be harder to enforce strong control access policies.

5.3. Attacks performed from the Tenant's Users Plane

The SFC Tenant's User Plane is not expected to have fine access control policies on the packets sent or received by users. Unless they are filtered, all packets are good candidates to the SFC Classifier. This provides the user some opportunities to test the behavior of the SFC.

In addition, the Tenant's Users Plane is not controlled by the SFC Tenant, and users may initiate communications where both ends - the client and the server - are under the control of the same user. Such communications may be seen as user controlled communications (UCC).

UCC may enable any user to monitor and measure the health of the SFC. This may be a useful information to infer information on the tenant's activity or to define when a DoS attack may cause more damage. One way to measure the health or load of the tenant's SFC is to regularly send a packet and measure the time it takes to be received, in order to estimate the processing time within the SFC.

UCC may enable any user to test the consistency of the SFC. One example of inconsistency could be that SFC decapsulation is not performed - or inconsistently performed - before leaving the SFC, which could leak some metadata with private information. For example, a user may send a spoofed packet. Suppose for example, that a request HTTP GET video.example.com/movie is received with some extra header information such as CLIENT_ID: 1234567890, or CLIENT_EMAIL: client@example.foo. If these pieces of information are derived from the source IP address, the attacker may collect them by changing the IP address for example. In this case, the spoofed packets as used to collect private and confidential information of the tenant's users. Note that such threat is not specific to SFC, and results from the combination of spoofed IP and non-authenticated IP address are used to identify a user. What is specific to SFC is that metadata are likely to carry multiple pieces of information - potentially non-authenticated - associated to the user. In the case above, meta-data is carried over the HTTP header. Inserting the metadata in the HTTP

header may be performed by a SF that takes its input from the SFC encapsulation. In addition, SFC encapsulation may also leak this information directly to a malicious node if that node belongs to the SFC plane. In this later case, the user builds on the top of and intrusion to the SFC Data Plane that is detailed later.

In some case, spoofed packet may impersonate other's tenants. Suppose for example that the same infrastructure is used by multi tenants, and which are identified by the IP address of their users. In this case, spoofing an IP address associated to another tenant may be sufficient to collect the information confidential and private information. The best current practice to prevent such leaks are usually ingress filtering for example, which prevents unlegitimate flows to enter the network. Note that ingress filtering may also be performed at higher layers such as at application layers to prevent unexpected applications to enter the network. When possible, the cost needs to be balanced with the risk by the SFC tenants.

Similarly, UCC may enable any user to infer packet has been dropped or is in a loop. Suppose a user send a spoofed packet and receives no response. The attacker may infer that the packet has been dropped or is in a loop. A loop is expect to load the system and sending a "well known packet" over the UCC and measuring the response time may determine whether the packet has been dropped or is in a loop.

Correlation of time measurement and spoofed packet over a UCC may provide various type information that could be used by an attacker.

- The attacker may correlate spoofed packet and time measurement in order discover the SFC topology or the logic of the SFC Classifier. Typically, it may infer when new SFs are placed in the SFC for example. In addition, as metadata are placed in band, the time response may also provide an indication of the size of the metadata associated to the packet. The combination of these pieces of information may help an attacker to orchestrate a future attack on a specific SF either to maximize the damages or to collect some metadata - like identification credentials.
- The attacker may also define the type of packets that require the SFC the more processing. Additional processing may be due a large set of additional metadata that require fragmentation, some packets that are not treated in a coherent and consistent manner within the SFC. Such information may be used for example to optimize a DoS attack. In addition, it could also be used in order to artificially increase the necessary resource of the Tenant in order to increase the cost of operation for running its service.

Time measurement and spoofed packet in combination with variable query rate over a UCC may provide information on the orchestration of the SFC itself. For example, the user may be able to detect when elasticity mechanisms are triggered. Such attack is not SFC specific, and may have occurred with traditional cloud mechanisms. However, the main difference between SFC and traditional cloud mechanisms is that SFC is a standard way to interconnect SF. In that sense, the use of SFC provides more details to the attack as non standard mechanisms.

An attacker may be able to leverage the knowledge that SFC is in use by specific carriers to effect the processing of data using the SFC system as a processor in the attack. This leads to a number of potential weaknesses in the Internet ecosystem.

An attacker may be able to characterize the type of client platforms using a web site by carefully crafting data streams that will be modified by the SFC system versus client systems that would view web data unmodified. For example, leveraging SFC and carefully crafted data, a malicious web site operator may be able to create a particularly formatted common file that when modified by a cellular operator for bandwidth savings creates a file that may crash, (creating a DoS attack) on a select set of clients. Clients not accessing that web site using the same RSFP would not experience any issues. Additionally, external examination of the malicious site would not demonstrate any malicious content, relying on the SF to modify the content.

A well crafted site could potentially leverage the variances of functionality from different RSFPs in order to GEO locate a user. An example would be creating an image file which when recompressed creates image artifacts rendering the image unusable, but allowing the user to respond to such an event, thereby letting the web site operator know the user has potentially moved from a higher to lower bandwidth network location within the area of a specific network operator.

5.4. Attacks performed from the SFC Data Plane

This section considers an attacker has been able to take control of an SFC component. As a result, the attacker may become able to modify the traffic and perform, on-path attacks, it may also be able to generate traffic, or redirect traffic to perform some kind of Man-in-the-middle attacks. This is clearly a fault, and security policies should be set to avoid this situation. This section analyses in case this intrusion occurs, the potential consequences on the SFC. As mentioned earlier, this section assumes all these actions are performed by an attacker. However, what is designated by

an attack may also result from misconfigurations at various layer. A SF or a SFF may become inadvertently configured or programmed which may result in similar outcomes as an attack. Whatever result in what we designate as an attack, the purpose of security requirements will be to detect, to analyse and mitigate such security breaches.

The traffic within the SFC Data Plane is composed of multiple layers. The traffic is composed of communications between SFC components. The transport between the SFC component is the transport protocol and is not considered in the SFC. It can typically be a L2 transport layer, or an L3 transport layer using various encapsulation techniques (vLAN, VxLAN, GRE, IPsec tunnels for example). Each of these transport layer adds or remove attack vectors. The transport layer carries SFC Encapsulated that are composed of an SFC Encapsulation envelope that carries metadata and a SFC payload that is the actual packet exchanged between the two end points.

As a result, attacker may use the traffic to perform attacks at various layers. More specifically, attacks may be performed at the transport layer, the SFC Encapsulation layer or the SFC payload layer.

- Attacks performed at the transport layer may be related to SFC in the sense that illegitimate SFC traffic could be provided to the SF. Typically, a malicious node that is not expected to communicate with that SF may inject packets into the SFC, such malicious node may eventually spoof the IP address of legitimate SF, so the receiving SF may not be able to detect the packet is not legitimate. Threats related to IP spoofing are described in [RFC6959] and may be addresses by authenticated traffic (e.g. using IPsec). Such threats are not related to SFC even though they may impact a given SF.
- the SFC Encapsulation as well as the SFC payload are usually considered as input by a SF. As such they may represent efficient vector of attacks for the SF. Attacks performed through SFC payload are similar as the ones described in the Tenant's Users Data Plane section. As a result, such attacks are not considered in this section, and this section mostly considers attacks based on the SFC Encapsulation and malicious metadata.

When an attacker is within the SFC Data Plane, it may have a full or partial control of one SF component in which case, the attacker is likely to compromise the associated SFCs. It could for example, modify the expected operation of the SFC. Note that in this case, the SFC may be appropriately provisioned and set, however, the SFC

does not operate as expected this may only be detected by monitoring and auditing the SFC Data Plane.

Although traffic authentication may be performed at various layers L2 L3 or at the SFC Encapsulation layer, this section considers the SFC traffic. As a result, the SFC traffic is authenticated if the SF is able to authenticate the incoming SFC packet.

When SFC traffic is not authenticated, an attacker may inject spoofed packet in any SFC component. The attacker may use spoofed packet to discover the logic of the SFC. On the other hand, the attacker may also inject packet in order to perform DoS attack via reflection. In fact, as NSH provides the ability to add metadata, some deployment may end up with payloads carrying large metadata. Addition of such overhead presents a vector for amplification within the SFC Data Plane and thus either load the network or the next SF. Note that amplification may be generated by metadata, the SFC payload, and the attacker may replay packets or completely craft new packets. In addition, the attacker may choose a spoofed packet to increase the CPU load on the SFC components. For example, it could insert additional metadata to generate fragmentation. Similarly, it may also insert unnecessary metadata that may need to be decapsulated and analyzed even though they may not be considered for further actions. Spoofed packet may not only be generated to attack the SFC component at the SFC layer. In fact spoofed packet may also target applications of the SF. For example an attacker may also forge packet for HTTP based application - like a L7 firewall - in order to perform a slowloris [SLOWLORIS] like attack. Note that in this case, such attacks are addressed in the Tenant's Users Data Plane section. The specificity here is that the attacker has a more advanced understanding of the processing of the SFC, and can thus be more efficient.

When SFC traffic is not authenticated, an attacker may also modify on-path the packet. By changing some metadata contained in the SFC Encapsulation, the attacker may test and discover the logic of the SFC. Similarly, when the attacker is aware of the logic of a SFC component, the attacker may modify some metadata in order to modify the expected operation of the SFC. Such example includes for example redirection to a SF which could result in overloading the SF and overall affect the complete SFC. Similarly, the attacker may also create loops within the SFC. Note that redirection may not occur only in a given SFC. In fact, the attacker may use SFC branching to affect other SFC. Another example would also include a redirection to a node owned by the attacker and which is completely outside the SFC. Motivation for such redirection would be that the attacker has full administrator privileges on that node, whereas it only has limited capabilities on the corrupted node. Such attack is a man-in-

the-middle attack. The important thing to note is that in this case the traffic is brought outside the legitimate SFC domain. In fact, performing a man-in-the-middle attack as described above means that the SFC domain has been extended. This can be easily performed in case all node of the data center or the tenant's virtual network is likely to host a SFC component. A similar scenario may also consider that the traffic could be redirected outside the data center or the tenant's virtual network if the routing of firewall rule enables such policies.

A direct consequence is that a corrupted SFC component may affect the whole SFC. This also means that the trust of a given SFC decreases with the number of SF involved as each SF presents a surface of attack.

An attacker may also perform passive attacks by listening to traffic exchanged throughout the SFC Data Plane. Such attacks are described in [RFC7258]. Metadata are associated to each packet. These metadata are additional pieces of information not carried in the packet and necessary for each SF to operate. As a result, metadata may contain private information such as identifiers or credentials. In addition, observing the traffic may provide information on the tenant's activity. Note that encryption only may not prevent such attacks, as activity may be inferred by the traffic load.

6. Security Requirements

This section aims at providing environment security requirements. These requirements are derived from the generalization of the threat analysis described in Section 5. More specifically, the threat analysis section was mostly illustrative, and its generalization leads us to the following requirements.

Although the security requirements are derived from described threats, the scope of security should be understood in a much broader way than addressing threats. In fact the primary purpose of the security requirements is to ensure the deployment of the SFC architecture can remain robust and stable.

The goal of this section is to provide some security requirements that should be checked against any evolution of the deployment of SFC architecture. The requirements should be understood and the risks of not following them should be evaluated with the current deployment as well as the foreseen evolutions.

Similarly, the document provides means to evaluate the consequences of a security breach, as well as means to detect them.

The motivations for the security requirements are:

- a) Preventing attacks
- b) Preventing misconfigurations - as far as stability and security of the SFC architecture is concerned.
- c) Providing means to evaluate the consequences of a security breach
- d) Making possible to audit, and detect any misbehavior that may affect stability and security of the SFC.

6.1. Plane Isolation Requirements

Plane Isolation consists in limiting the surface of attack of the SFC Data Plane by controlling the interfaces between the SFC Data Plane and the other planes.

Complete isolation of the planes is not possible, as there are still some communications that must be enabled in order to benefit from the benefits of SFC. Typically the SFC Control Plane configures the SFC elements used by the SFC Data Plane. Similarly, access to the SFC Control Plane may be performed remotely, in which case interaction between the SFC Tenant's User and the SFC Control Plane may be considered. As a result, isolation should be understood as enabling communications between planes in a controlled way.

This section lists the recommendations so communication between planes can be controlled. This involves controlling communications between planes as well as controlling communication within a plane.

The requirements listed below applies to all planes, whereas the following subsection are more specific to each plane, providing recommendations on the interface with the SFC Data Plane.

REQ1: In order to increase isolation every plane that communicates with another plane SHOULD use a dedicated interface. In our case, the SFC Management Plane, the SFC Control Plane and the SFC Data Plane SHOULD use dedicated networks and dedicated interfaces. Isolation of inter-plane communication may be enforced using different ways. How isolation is enforced depends on the type of traffic, the network environment for example, and within a given SFC architecture different techniques may be used for the different planes. One way to isolate communications is to use completely different network on dedicated NICS. On the other hand, depending on the required level of isolation, a logical isolation may be performed using different IP addresses or ports with network

logically isolated - that is using for example different VxLAN, or GRE tunnels. In this case, isolation relies on the trust associated to the different switches and router. In case of a lack of trust on the on-path elements, authenticated encryption may be used to provide a logical isolation. With authenticated encryption, trust is placed on the end points. Note also that encryption can also be used in combination of other isolation mechanisms in order to increase the level of isolation.

- REQ2: Activity between planes SHOULD be monitored and regularly audited. At least operations performed between the planes as well as the source and destination should be logged. When possible the identity of the identities should also be logged. Activity may be performed independently by the different planes as well as by different actors such as the SFC Tenants, the infrastructure provider. The level of information available may also differ between planes and actors.
- REQ3: Traffic and communications between planes SHOULD be filtered traffic or rate-limited. Filtering and rate-limiting policies may be finer grained and may apply for a subset of traffic.

The above requirements mostly corresponds to the architecture best current practice. Isolation is mostly motivated to avoid the planes to interact on each other. For example the load on the SFC Data Plane should not affect the SFC Control Plane and SFC Management Plane communications. Such requirements are also current best practices.

Such recommendations are thus strongly recommended even in the case the two planes are considered to belong to trusted environments.

6.1.1.1. SFC Control Plane Isolation

In order to limit the risks of an attack from the SFC Control Plane, effort should be made in order to restrict the capabilities and the information provided by the SFC Data Plane to the SFC Control Plane to the authorized tenants only. In this case the authorized tenants are the users or organizations responsible for the SFC domain.

- REQ4: Tenants of the SFC Control Plane SHOULD authenticate in order to prevent tenant's usurpation or communication hijacking.
- REQ5: Communications between SFC Control Plane and the SFC Data Plane MUST be authenticated and encrypted in order to preserve privacy. The purpose of encryption in this case prevents an attacker to be aware of the action performed by the SFC

Control Plane. Such information may be used to orchestrate an attack - especially when SFC component report their CPU/network load.

REQ6: Strong access control policies SHOULD be enforced. Control SHOULD be performed on the engaged resource (e.g. CPU, memory, disk access for example) and SHOULD be associated explicitly to authorized tenants. By default, a tenant SHOULD be denied any access to resource, and access SHOULD be explicit.

Given the SFC Control Plane traffic load that is expected to be light - at least compared to the SFC Tenant's Users Data Plane or the SFC Data Plane. As a result, encryption is not expect to impact the performances of the SFC architecture. Given the effort to migrate from an non authenticated (and non protected) communications to a protected communication, we recommend these requirements to be considered even in trusted environments. By protecting these communications by design, the deployed SFC architecture is also ready for future expansion of the Control Plane outside the initial trusted domain. This could typically includes the evolution to multiple tenants as well as the inclusion of tenants that remotely access the SFC Control Plane.

Access Control policies can be enforced in various ways. One way could be to consider the systems of the SF to limit the resources associated to each tenants. Other ways include the use of API in order to limit the scope of possible interactions between the SFC Control Plane and the SFC Data Plane. This is one way to limit the possibilities of the tenants. In addition, each of these actions should be associated an authorized tenant, as well as authorized parameters. The use of API belongs to best practices and so is strongly recommended even in trusted environments.

REQ7: Audit SHOULD be performed regularly to check access control policies are still up-to-date and prevent non-authorized users to control the SFC Data Plane.

The purpose of audits is to provide evidences when something went wrong. As a result, audit facilities are expected to be provided even in trusted environments.

6.1.2. SFC Management Plane Isolation

The requirements for the SFC Control Plane and SFC Management Plane are similar. The main difference of the interfaces between the SFC Management Plane and the SFC Control Plane is that it is less likely that APIs could be used to configure the different SFC components.

As a result, users of the SFC Management Plane are likely to have a broader and wider control over the SFC component.

REQ8: it is RECOMMENDED to enforce stronger authentication mechanisms (for example relying on hardware tokens or keys) and to limit the scope of administrative roles on a per component basis.

REQ9: SFC Control Plane and SFC Management Plane may present some overlap. Each SFC component MUST have clear policies in case these two planes enter in conflict.

6.1.3. Tenant's Users Data Plane Isolation

The Tenant's Users Data Plane is supposed to have less restricted access control than the other SFC Management Plane and SFC Control Planes. A typical use case could be that each tenant are controlling and managing the SFC in order to provide services to their associated users. The number of users interacting with the SFC Data Plane is expected to be larger than the number of tenants interacting with the SFC Control and SFC Management Planes. In addition, the scope of communications initiated or terminating at the user end points is likely to be unlimited compared to the scope of communications between the tenants and the SFC Control Plane or SFC Management Plane. In such cases, the tenant may be provided two roles. One to grant access to the SFC, and another one to control and manage the SFC. These two roles should be able to interact and communicate.

REQ10: Users SHOULD be authenticated, and only being granted access to the SFC if authorized. Authorization may be provided by the SFC itself or outside the SFC.

REQ11: Filtering policies SHOULD prevent access to a user, or traffic when a malicious behavior is noticed. A malicious activity may be noticed once a given behavioral pattern is detected or when unexpected load is monitored in the SFC Data Plane.

REQ12: Tenant's User Plane SHOULD be monitored, in order to detect malicious behaviors.

REQ13: When SFC is used by multiple tenants, each tenant's traffic SHOULD be isolated based on authenticated information. More specifically, the use of a Classifier that can easily be spoofed like an IP address SHOULD NOT be used.

REQ14: It is RECOMMENDED that user's access authorization be performed outside the SFC. In fact granting access and treating the traffic are two different functions, and we

RECOMMEND they remain separated. Then, splitting these two functions makes it possible for a tester to perform tests of an potential attacker, without any contextual information. More specifically, having a traffic identified as associated to test by the SFC reduces the scope of the tests simply because an attacker will not be considered as a tester. For that reason, we RECOMMEND authorization is performed outside the SFC, and SFC deployment may not be designed to authenticate end users.

The remaining requirements are associated to monitoring the network and providing interactions between the access and the SFC. This interaction may be provided outside SFC itself.

6.2. SFC Data Plane Requirements

This section provides requirements and recommendation for the SFC Data Plane.

- REQ15: Communications within the SFC Data Plane SHOULD be authenticated in order to prevent the traffic to be modified or injected by an attacker. As a result, authentication includes the SFC Encapsulation as well as the SFC payload.
- REQ16: Communication MUST NOT reveal privacy sensitive metadata.
- REQ17: The metadata provided in the communication MUST be limited in in term of volume as to limit the amplification factor as well as fragmentation.
- REQ18: Metadata SHOULD NOT be considered by the SFF for forwarding decision. In fact, the inputs considered for switching the packet to the next SFF or a SF should involve a minimum processing operation to be read. More specifically, these inputs are expected fixed length value fields in the SFC Encapsulation header rather than any TLV format.
- REQ19: When multiple tenants share a given infrastructure, the traffic associated to each tenant MUST be authenticated and respective Tenant's Users Planes MUST remain isolated. More specifically, if for example, a SFC Classifier is shared between multiple tenants. The Classifier used to associate the SFC MUST be authenticated. This is to limit the use of spoofed Classifiers. In any case, the SFC component that receives traffic from multiple tenants is assumed to be trusted.

REQ20: Being a member of a SFC domain SHOULD be explicitly mentioned by the node and means should be provided so the SFC domain the node belongs to may be checked. Such requirement intends to prevent a packet to go outside a SFC domain, for example in the case of a man-in-the-middle attacks, where a redirection occurs outside the SFC domain. It is expected that most deployment will rely on border / port mechanisms that prevent outsider users from injecting packets with spoofed metadata. Although such mechanisms are strongly recommended to deploy, in case of failure, they do not prevent man-in-the-middle attack outside the SFC domain.

Authentication of the traffic within the SFC Data Plane is particularly recommended in an open environment where third party SF or SFF are involved. It can also be recommended when a strong isolation of the traffic is crucial for the infrastructure or to meet some level of certification. In addition, authentication may also be performed using various techniques. The whole packet may be authenticated or limited to some parts (like the flow ID). Authenticating the traffic and how or what to authenticate is a trade off between the risk associated and the cost of encryption. When possible we recommend to authenticate, but we also consider that the price may be too high in controlled and small trusted environment.

Metadata is an important part of the SFC architecture, and their impact on security should be closely evaluated. It is the responsibility of the SFC administrator to evaluate the privacy associated by the metadata - section 5.2.2 of [RFC6973] - and according to this evaluation to consider appropriated mechanisms to prevent the privacy leakage. Mechanisms should be provided even though they may not be activated.

As a general guidance exposing privacy sensitive metadata in any communications between two any SFC component should be avoided. [One way, for example to avoid exposing privacy sensitive metadata is to include a reference to the metadata instead of the metadata itself. Another way could be to encrypt the metadata itself - but that is part of the solution space.] Applying this principle prevents any private oriented data to be leaked. This requirement is mandatory when the SFC is not deployed in a trusted environment.

When exposition of the privacy sensitive metadata cannot be avoided and you are in a trusted domain, then exposing privacy sensitive metadata may be considered as long as they do not leak outside the boundaries of the trusted environment. In this case, the security is delegated to the security policies of the trusted environment boundaries, that may be outside the scope of SFC. More especially, the security policies may be for example enforced by a firewall. In

this specific case, the trusted environment MUST prevent leakage of the metadata out of the trusted environment and MUST ensure that untrusted node cannot access in any way the communications within the trusted environment.

The reason this requirement is set to MUST is to specify that if one does not follow the requirement it is at its own risk and must provide the necessary means to prevent any leak - in our case enforcing the necessary security policies that your environment / deployment needs.

Similarly, it is the responsibility of the administrator to define what an acceptable size for metadata is. Even in trusted environment, we recommend the SFC administrator be able to define and change this level.

Processing metadata by the SFF seems also expensive, and it is the responsibility of the SFC administrator to evaluate whether processing metadata by the SFF may impact the SFC architecture. In addition, metadata are expected to be associated to SF as opposed to the forwarding information that are associated to the SFF. These inputs have different functions, are associated to different processing rules, and may be administrated by different parties. It is thus part of the general good practise to split these functionalities. Optimization may require to combine the analysis of metadata and forwarding information, but this should be handled cautiously.

Assertion of belonging to a security domain, is especially recommended in open environments. This may also partly be addressed by node authenticating.

In addition, the following operational requirements have been identified:

REQ21: SFC components SHOULD be uniquely identified and have their own cryptographic material. In other words the use of a shared secret for all nodes SHOULD NOT be considered as one corrupted node would be able to impersonate any node of the SFC Data Plane. This is especially useful for audit.

REQ22: Activity in the SFC Data Plane MUST be monitored and audited regularly. Audit and log analysis is especially useful to check that SFC architecture assessments. They can be useful to detect a security breach for example before it is being discovered and exploited by a malicious user. Monitoring the system is also complementary in order to provide alarms when a suspicious activity is detected. Monitoring enables the

system to react to unexpected behaviors in a dynamic way. Both activities are complementary as monitoring enables to counter suspicious behavior and audit may detect misconfiguration or deep causes of a malicious behavior. For these reasons, audit and monitoring facilities are expected even in trusted environment.

REQ23: Isolate the Plane with border and firewall to restrict access of SFC components to legitimate traffic. More specifically, SFC components are supposed to be accessed only via dedicated interfaces. Outside these interfaces, inbound or outbound traffic SHOULD be rejected.

6.3. Additional Requirements

REQ24: SFC Encapsulation SHOULD carry some identification so it can be associated to the appropriated SFP as well as its position within the SFC or SFP. Indicating the SFP ID may be sufficient as long as a SFP can uniquely be associated to a single SFC. Otherwise, the SFC should be also somehow indicated. This is especially useful for audit and to avoid traffic coming from one SFC to mix with another SFC. Authentication of the SFP ID is one way to enforce SFP ID uniqueness. This may not be mandatory, but large deployment or deployment that are involving multiple parties are expected enforce this. In fact assuming SFP ID will have no collision is an hypothesis that may be hard to fulfill over time.

REQ25: Although this requirement is implementation specific, it is RECOMMENDED that SFF and SF keep separate roles. SFF should be focused on SF forwarding. As a result, they are expected to access a limited information from the packet - mostly fixed size information. SF on the other hand are service oriented, and are likely to access all SFC information which includes metadata for example. The reasons to keep these functions are clearly different and may involve different entities. For example, SF management or SF configuration may involve different administrators as those orchestrating the SFC.

REQ26: SFC Encapsulation SHOULD be integrity protected to prevent attackers from modifying the SFP ID. See Data Plane communication Requirements and considerations)

7. Security Considerations

8. Privacy Considerations

9. IANA Considerations

10. Acknowledgments

The authors would like to thank Joel Halpern, Mohamed Boucadair and Linda Dunbar for their valuable comments. Similarly the authors would also like to thank Martin Stiemerling for its careful review as well as its recommendations.

11. References

11.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <<http://www.rfc-editor.org/info/rfc4949>>.
- [RFC6959] McPherson, D., Baker, F., and J. Halpern, "Source Address Validation Improvement (SAVI) Threat Scope", RFC 6959, DOI 10.17487/RFC6959, May 2013, <<http://www.rfc-editor.org/info/rfc6959>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<http://www.rfc-editor.org/info/rfc6973>>.
- [RFC7258] Farrell, S. and H. Tschofenig, "Pervasive Monitoring Is an Attack", BCP 188, RFC 7258, DOI 10.17487/RFC7258, May 2014, <<http://www.rfc-editor.org/info/rfc7258>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.

11.2. Informative References

[I-D.ietf-sfc-nsh]

Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-01 (work in progress), July 2015.

[I-D.ietf-sfc-architecture]

Halpern, J. and C. Pignataro, "Service Function Chaining (SFC) Architecture", draft-ietf-sfc-architecture-11 (work in progress), July 2015.

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

Li, H., Wu, Q., Huang, O., Boucadair, M., Jacquenet, C., Haeffner, W., Lee, S., Parker, R., Dunbar, L., Malis, A., Halpern, J., Reddy, T., and P. Patil, "Service Function Chaining (SFC) Control Plane Components & Requirements", draft-ietf-sfc-control-plane-00 (work in progress), August 2015.

[SLOWLORIS]

Wikipedia, "Slowloris", <https://en.wikipedia.org/wiki/Slowloris_%28software%29>.

Authors' Addresses

Daniel Migault (editor)
Ericsson
8400 boulevard Decarie
Montreal, QC H4P 2N2
Canada

Phone: +1 514-452-2160
Email: daniel.migault@ericsson.com

Carlos Pignataro
Cisco Systems, Inc.
7200-12 Kit Creek Road
Research Triangle Park, NC 27709
USA

Phone: +1 919-392-7428
Email: cpignata@cisco.com

Tirumaleswar Reddy
Cisco Systems, Inc.
Cessna Business Park, Varthur Hobli
Bangalore, Karnataka 560103
India

Phone: +91 9886
Email: tiredy@cisco.com

Christopher Inacio
CERT, Software Engineering Institute, Carnegie Mellon University
4500 5th Ave
Pittsburgh, PA 15213
USA

Phone: +1 412-268-3098
Email: inacio@cert.org

Network Working Group
Internet-Draft
Intended status: Informational
Expires: May 1, 2017

B. Sarikaya
Huawei
M. Boucadair
Orange
D. von Hugo
Telekom Innovation Laboratories
October 28, 2016

Service Function Chaining Metadata Type 1 and Type 2
draft-sarikaya-sfc-metadatat1t2-00.txt

Abstract

With the definition of service function chain data plane protocol there comes the need to define the context data needed in the service function chain use cases. This document gives an account of all context data defined so far as Network Service Header metadata Type 1 and Type 2 context headers. Next, the document discusses the various options that can be taken in standardizing service function chain metadata.

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 1, 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
 2. Context Metadata Definitions 2
 3. Processing Metadata Type 1 and Type 2 5
 4. IANA Considerations 6
 5. Security Considerations 6
 6. Acknowledgements 6
 7. References 6
 7.1. Normative References 6
 7.2. Informative References 7
 Authors' Addresses 8

1. Introduction

Network Service Header (NSH) [I-D.ietf-sfc-nsh] is the Service Function Chaining (SFC) data plane protocol. The SFC architecture is defined in [RFC7665].

NSH has the function of carrying context data in the form of context header. NSH metadata Type 1 is composed of a 4-byte base header, 4-byte service path header. It contains four mandatory Context Headers, 4-byte each. For additional metadata that needs to be carried, NSH metadata type 2 is defined. Type 2 metadata is composed of a 4-byte base header carrying Type value of 0x02, 4-byte service path header followed by variable length context headers in the form of type-length-value or TLV.

Many context headers were proposed by many documents. In this document we survey existing drafts that propose new context metadata and then discuss different options that can be taken to standardize this work.

The reader should be familiar with the terms defined in [RFC7665] and [I-D.ietf-sfc-nsh].

2. Context Metadata Definitions

[I-D.quinn-sfc-nsh-tlv] defines NSH metadata Type 2 TLVs such as forwarding context, subscriber/user info, tenant, application ID, content type, ingress network information, flow ID, source and/or destination groups, universal resource identifier (URI).

Some of these TLVs are defined in other documents, like App ID, Context ID in [I-D.napper-sfc-nsh-broadband-allocation]. Also for Application ID, even though the document references [I-D.penno-sfc-appid], [I-D.penno-sfc-appid] seems to mean Classification Engine ID and Selector ID for the Application ID.

The purpose of [I-D.quinn-sfc-nsh-tlv] is to document syntactic structure of the TLVs. No other additional information about the metadata processing is within the scope of this document. The document mentions no use cases in which the TLVs defined are needed. An implementer will need to refer to other documents to understand the exact behavior for handling those contexts.

[I-D.napper-sfc-nsh-broadband-allocation] supports use cases in [I-D.ietf-sfc-use-case-mobility].

This document defines meta data Type 1 with endpoint ID, e.g. for IMSI or MSISDN or wireline subscriber ID with 64-bit length. It also defines ServiceTag to identify that the Service Information field contains information related to the Access Network (AN) for the subscriber. Service information could contain IP-CAN type, QoS class, congestion level, etc. for a 3GPP Radio Access Network (RAN). Context ID field allows the subscriber/endpoint ID field to be scoped. Context ID contains the incoming VRF, VxLAN VNID, VLAN, or policy identifier within which the Subscriber/Endpoint ID field is defined.

In addition, the document defines a meta data Type 2 TLV to be associated with 3GPP registry. The intent here is to offer this TLV for the use of 3GPP to extend the meta data to meet the needs of 3GPP use cases. However, it was not stated if 3GPP requested such an allocation.

[I-D.wang-sfc-nsh-ns-allocation] addresses the use cases for network security defined in [I-D.wang-sfc-ns-use-cases].

It defines a recommended security context allocation as a meta data Type 1 TLV. It is intended to define session ID, tenant ID, destination/ source class for the logical classification of the destination/ source of the traffic, destination/ source score which contains security classification results for communicating immediate actions and accumulated verdicts to downstream Service Functions.

[I-D.wang-sfc-nsh-ns-allocation] also mentions that the security context allocation, although defined as Type 1, it may also form a MD-Type 2 metadata TLV, possibly implying that the sizes of data such as session/ tenant ID, etc. may need to become longer. As a result, they may need to become variable length data as in Type 2 meta data

TLVs. This document defines network security allocation specifics, basically explaining the semantics of the metadata they define in the document.

[I-D.sarikaya-sfc-hostid-serviceheader] addresses use cases that require revealing host and/ or subscriber related information to upstream SFs as well as extreme low latency service and ultra-high reliability applications use cases.

From the analysed use cases, there comes the need to come up with definition of host, subscriber, slice identifier and service identifier SFC meta data Type 2 TLVs. Apart from defining these TLVs, the document gives details of post processing in various nodes such as ingress/egress border nodes, SFC-aware Service Functions and Proxies. Such post processing is defined as normative behavior. Since host and subscriber identifiers may reveal private information about the host and/or the subscriber, the document also defines normative behavior needed to protect the privacy of the hosts and subscribers in an operator network.

[I-D.sarikaya-sfc-hostid-serviceheader] is unique among the documents discussed in this document because it defines the post processing normative behavior related to the host and subscriber identifier meta data Type 2 TLVs. Also the use cases are defined in the same document not as a separate document as in the other cases.

[I-D.penno-sfc-packet] addresses the problem of sending packets in the reverse direction to the source of the current in-process packet/ flow. It defines SF Reverse Packet Request as Type 1 metadata TLV. This is defined as Version 1 (as opposed to Version 0 of NSH MD-type 1 in [I-D.ietf-sfc-nsh]) with OAM Protocol replacing the next protocol field and with Reverse Packet Request added to the end of mandatory context header octets for SFC as an additional 4-octet for OAM.

This document also proposes 5 new metadata on service-path invariants, service-path default, bidirectional clonable, unidirectional clonable and service-function-mastered metadata. Their structure specifics are not specified.

[I-D.penno-sfc-packet] gives a detailed explanation of the use of the metadata defined, all the semantic information, pre and post processing details at various nodes.

[I-D.meng-sfc-nsh-broadband-allocation] defines Type 1 metadata called Broadband Context Allocation support service function chaining in a broadband service provider network. It defines Source Node, Source Interface, User and VLAN IDs.

[I-D.vallamkonda-sfc-metadata-model] does not define any Type 1 or Type 2 meta data TLVs, viewing such meta data as conveying preprocessing information about the packet, this document attempts to formally define the post processing information. To that end, it defines a vocabulary and information model for metadata. The document gives metadata information model example definitions for routing domain, IP endpoint, flow and traffic policy indication.

3. Processing Metadata Type 1 and Type 2

Some options are discussed below for processing NSH TLVs:

1. List the structure of meta data in one single document as a registry. The document is not supposed to contain any post processing information. [I-D.quinn-sfc-nsh-tlv] attempts this choice for some Type 2 TLVs. Currently there is no such document for Type 1 TLVs. Note that in the case of keeping a registry document, it is not clear how the post processing behavior (normative or optional) will be specified for the TLVs. One option is to keep such information in separate document. If such a strategy is adopted then the advantages obtained from documenting all TLVs in one document disappears because the implementers would need to consult many documents instead of only one.
2. All documents defining new meta data Type 1 and Type 2 TLVs are treated individually for standardization. This approach has the advantage of keeping all meta data Type 1 and Type 2 TLVs in separate and dedicated documents together with all the information that the implementers may need. This could be a strong positive especially if we consider the fact that the meta data are being defined for very many use cases and scenarios. It is unlikely that one implementer would need to implement a large number of these TLVs, thereby defeating the need for combining them in a single document.
3. Together with choice 1 above, while combining all TLVs in one document, it could be possible to keep post processing information related to the meta data can be considered individually for standardization.
4. Together with choice 2 above, Type 1 TLVs can be combined in one document but all Type 2 TLVs can be considered individually in separate dedicated documents.

A document intended to keep a registry of all TLVs can be an informational document. Companion documents defining semantics of

Type 1 and Type 2 metadata needs to be standard track in order to take the recommendations on processing the data into effect.

Another issue is the importance of Type 1 metadata and Type 2 metadata. It seems to be difficult to argue that Type 1 metadata is more important. The metadata defined in [I-D.wang-sfc-nsh-ns-allocation] is a good example as it can be defined either as Type 1 or Type 2. The same considerations could possibly be made for other documents.

It is recommended that the metadata defined be given serious consideration as to the merit of the use case that needs the metadata to the Service Function Chaining rather than syntactic considerations of Type 1 or Type 2.

4. IANA Considerations

None.

5. Security Considerations

This document does not introduce any security issues.

6. Acknowledgements

TBD.

7. References

7.1. Normative References

- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

7.2. Informative References

- [I-D.ietf-sfc-use-case-mobility]
Haefner, W., Napper, J., Stiemerling, M., Lopez, D., and J. Uttaro, "Service Function Chaining Use Cases in Mobile Networks", draft-ietf-sfc-use-case-mobility-07 (work in progress), October 2016.
- [I-D.liu-sfc-use-cases]
Will, W., Li, H., Huang, O., Boucadair, M., Leymann, N., Qiao, F., Qiong, Q., Pham, C., Huang, C., Zhu, J., and P. He, "Service Function Chaining (SFC) General Use Cases", draft-liu-sfc-use-cases-08 (work in progress), September 2014.
- [I-D.meng-sfc-nsh-broadband-allocation]
Meng, W. and C. Wang, "NSH Context Header - Broadband", draft-meng-sfc-nsh-broadband-allocation-01 (work in progress), May 2016.
- [I-D.napper-sfc-nsh-broadband-allocation]
Napper, J., Surendra, S., Muley, P., and W. Henderickx, "NSH Context Header Allocation -- Broadband", draft-napper-sfc-nsh-broadband-allocation-01 (work in progress), October 2016.
- [I-D.penno-sfc-appid]
Penno, R., Claise, B., Pignataro, C., and C. Fontaine, "Using Application Identification in Services Function Chaining Metadata", draft-penno-sfc-appid-05 (work in progress), August 2016.
- [I-D.penno-sfc-packet]
Penno, R., Pignataro, C., Yen, C., Wang, E., Leung, K., and D. Dolson, "Packet Generation in Service Function Chains", draft-penno-sfc-packet-03 (work in progress), April 2016.
- [I-D.quinn-sfc-nsh-tlv]
Quinn, P., Elzur, U., Majee, S., and J. Halpern, "Network Service Header TLVs", draft-quinn-sfc-nsh-tlv-02 (work in progress), October 2016.
- [I-D.sarikaya-sfc-hostid-serviceheader]
Boucadair, M., Hugo, D., and B. Sarikaya, "Service Function Chaining Service, Subscriber and Host Identification Use Cases and Metadata", draft-sarikaya-sfc-hostid-serviceheader-03 (work in progress), July 2016.

[I-D.vallamkonda-sfc-metadata-model]

sunilvk@f5.com, s., Dunbar, L., and D. Dolson, "A Framework for SFC Metadata", draft-vallamkonda-sfc-metadata-model-01 (work in progress), July 2016.

[I-D.wang-sfc-ns-use-cases]

Wang, E., Leung, K., Felix, J., and J. Iyer, "Service Function Chaining Use Cases for Network Security", draft-wang-sfc-ns-use-cases-02 (work in progress), October 2016.

[I-D.wang-sfc-nsh-ns-allocation]

Wang, E. and K. Leung, "Network Service Header (NSH) Context Header Allocation (Network Security)", draft-wang-sfc-nsh-ns-allocation-01 (work in progress), October 2016.

Authors' Addresses

Behcet Sarikaya
Huawei
5340 Legacy Dr.
Plano, TX 75024

Email: sarikaya@ieee.org

Mohamed Boucadair
Orange
Rennes 3500, France

Email: mohamed.boucadair@orange.com

Dirk von Hugo
Telekom Innovation Laboratories
Deutsche-Telekom-Allee 7
D-64295 Darmstadt
Germany

Email: Dirk.von-Hugo@telekom.de

Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: May 3, 2017

Vu Anh Vu
Younghan Kim
Soongsil University
October 30, 2016

Controlling Service Function Access to Network Service Header
draft-vu-sfc-sf-access-control-00

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 documents.

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

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

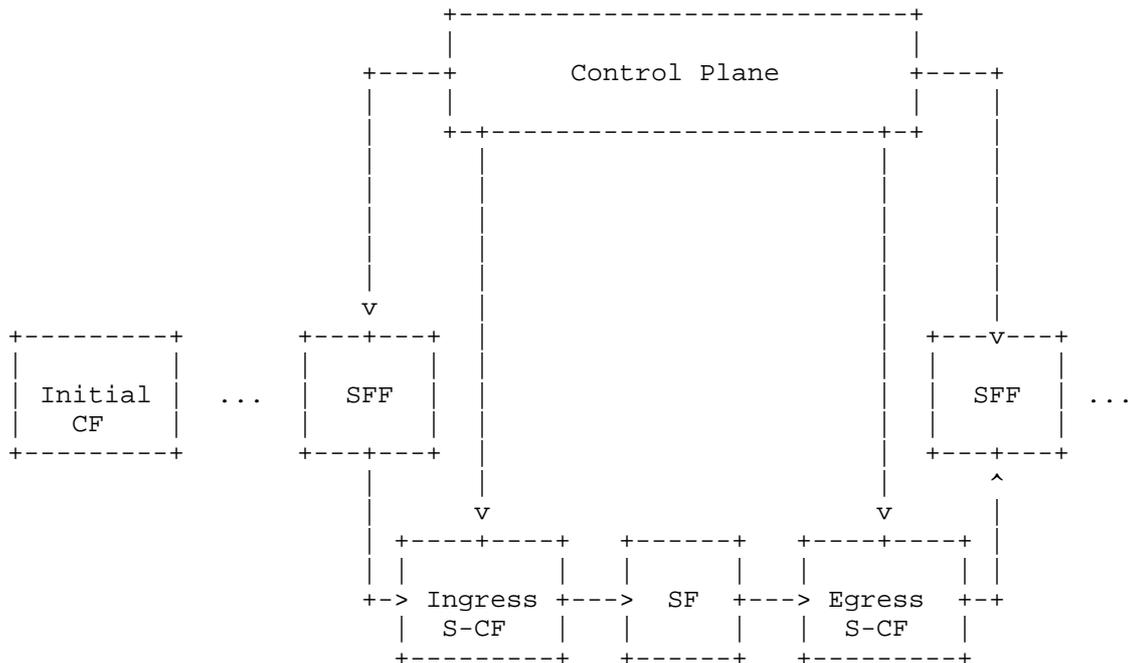


Figure 1: Controlling SFs access to NSH in SFC architecture

In this mechanism, the ingress and egress S-CF must exchange stored ACS data in either way:

- o Shared storage
- o 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 SFC corresponds with a packet it receives. Current solutions for this include:

- o 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)
- o 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.

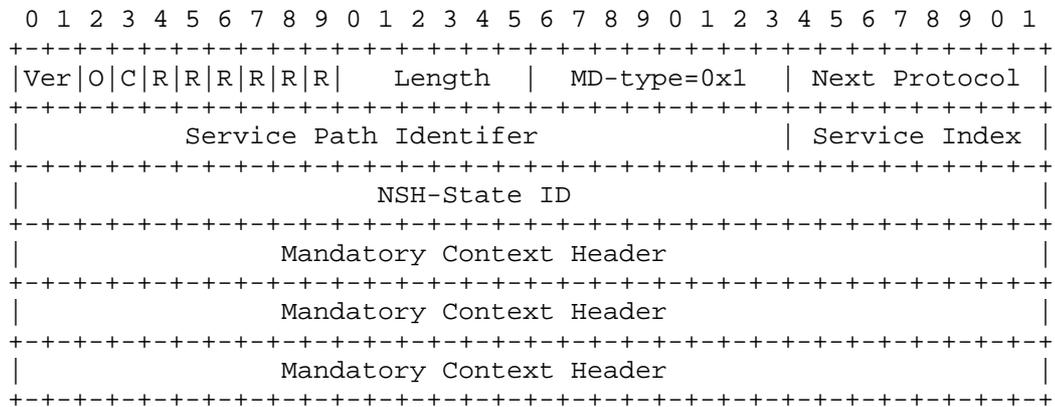


Figure 2: NSH-state ID in NSH Metadata type 1

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]

Boucadair, M., "Service Function Chaining (SFC) Control Plane Components & Requirements", draft-ietf-sfc-control-plane-08 (work in progress), October 2016.

- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

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

Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: April 28, 2017

E. Wang
K. Leung
J. Felix
J. Iyer
Cisco Systems Inc.
October 25, 2016

Service Function Chaining Use Cases for Network Security
draft-wang-sfc-ns-use-cases-02

Abstract

Enterprise networks deploy a variety of security devices to protect the network, hosts and endpoints. Network security devices, both hardware and virtual, operate at all OSI layers with scanning and analysis capabilities for application content. Multiple specific devices are often deployed together for breadth and depth of defense. This document describes use cases of Service Function Chaining (SFC) when deploying network security devices in the manner described above and also puts forth requirements for their effective operation.

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 April 23, 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 3
- 2. Definition Of Terms 3
- 3. Characteristics of Security Service Functions 4
- 4. Use Cases 5
 - 4.1. Service Classification Use Cases 5
 - 4.1.1. Service classification for bi-directional traffic 5
 - 4.1.2. Service Classifier to distinguish initiator and responder 6
 - 4.1.3. Service Classification based on network and application criteria 7
 - 4.1.4. Switching Service Function Paths based on inspection and scanning results 8
 - 4.2. Service Function Use Cases 10
 - 4.2.1. Service Classifier-capable Service Function 10
 - 4.2.2. Service Functions operating on L5 or L7 data 10
 - 4.2.3. Service Function mid-stream pick-up 10
 - 4.2.4. Bypassing a particular Service Function 11
 - 4.2.5. Receive-only Service Functions 13
 - 4.3. Service Data Handling Use Cases 13
 - 4.3.1. Service Function injected new packet 13
 - 4.3.2. Service Function initiated connections 14
 - 4.3.3. Security classification results 15
- 5. General Requirements 17
- 6. Security Considerations 18
- 7. Acknowledgments 18
- 8. IANA Considerations 18
- 9. References 18
 - 9.1. Normative References 19
 - 9.2. Informative References 19
- Authors' Addresses 19

1. Introduction

Network security service nodes participate in Service Function Chaining (SFC) to provide comprehensive solutions for securing campus and data center enterprise networks. Often, network operators deploy various types and instances of security service nodes. These nodes are complementary to one another for the purpose of coverage, depth of defense, scalability and availability.

In addition to packet forwarding, network security devices can buffer, inject or block certain packets, as well as proxy entire connections. Most of the network security devices maintain state at the connection, session or transaction levels. When used in a SFC environment these security Service Function actions and properties require careful design and extension including the Service Classifier and Service Function itself. This document attempts to describe the detailed use cases that lead to the requirements to support network security functions in SFC.

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 RFC 2119 [RFC2119].

2. Definition Of Terms

This document uses the terms as defined in RFC 7498 [RFC7498], RFC 665 [RFC7665] and [I-D.ietf-sfc-nsh].

In addition the following terms are defined.

Security Service Function (Security SF): A Security Service Function is a Service Function that carries out specific security tasks. We limit the scope of security functions to network security in this document (as opposed to functions such as endpoint security). In addition to the general forwarding action, a Security Service Function can buffer, proxy, inject or block certain packets based on its policy. A Security Service Function can maintain state at the connection, session or transaction levels. Sample Security Service Functions are: 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.

Flow: A flow is a uni-directional traffic stream identified by network layer attributes, specifically IP addresses and TCP/UDP ports for TCP/UDP traffic.

Connection: A connection is a bi-directional traffic stream composed of two flows sharing the same network layer attributes.

3. Characteristics of Security Service Functions

Most Security Service Functions are stateful. They maintain state at the connection, session or transaction levels, depending on the OSI layers that they act on. Many Security Functions require receiving both directions of the client-server traffic in order to maintain state properly. Asymmetric traffic must be normalized before packets reach the Security Functions.

Security Service Functions operate on network layer data with specific behaviors. For example:

1. A Firewall tracks TCP state between the TCP client and server. TCP packets that do not correspond to the Firewall's maintained state are likely to be dropped.
2. A Firewall can modify the L3/L4 headers for NAT [RFC3022]. The flow attributes in the packet header may be changed after the packet egresses the Firewall.
3. A Firewall can proxy a TCP connection by sending a TCP ACK on behalf of the endpoint. From the SFC perspective, this results in Service Function generated packets being injected into the service path in the reverse direction.
4. A Firewall or DDoS mitigator can inject TCP layer challenges to the originating client before the intended server receives a packet from the client.

Security Functions also handle packets and examine data at higher OSI layers. For example:

1. A Firewall can inspect the HTTP header and body data. Based on the inspection results, the firewall can decide to drop the packet and/or block the connection completely.
2. A Web proxy can inject an HTTP challenge page into an HTTP transaction for the purposes of authentication and identity collection.
3. At the enterprise edge, a TLS proxy, when authorized, operates as a trusted Man-in-the-Middle to proxy the TLS handshake and decrypt the packet data. The TCP payload may be completely different between ingress and egress of TLS Proxy.
4. A stream scanning service examines a certain set of application data. File scanning engines examine file streams of specific types.

4. Use Cases

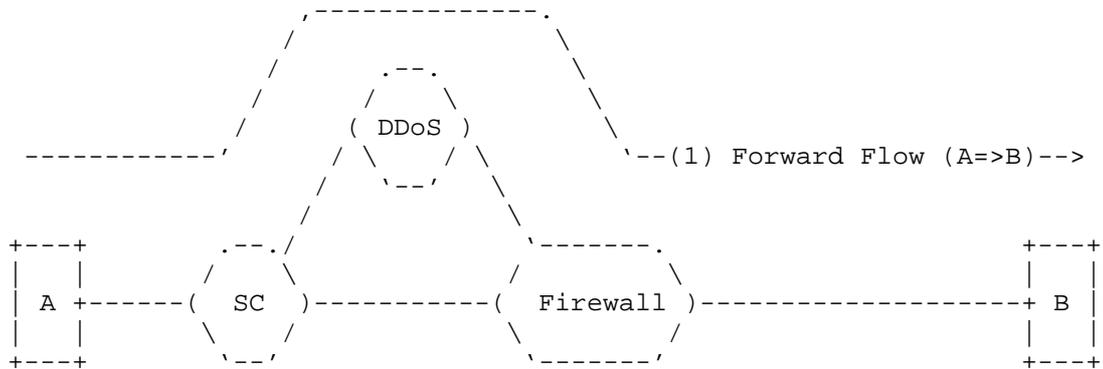
4.1. Service Classification Use Cases

4.1.1. Service classification for bi-directional traffic

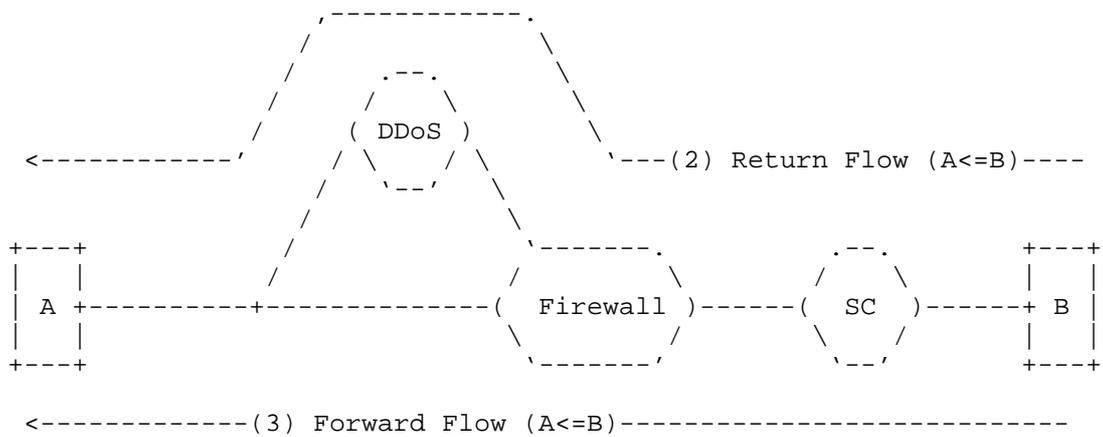
Many Security Service Functions require receiving bi-directional traffic of a connection. For example, a DDoS mitigator may require to see the return traffic to maintain proper state.

Return traffic (i.e. server to client response) should be classified based on the forward traffic (i.e. the client to server request). This allows server's return traffic to be associated with the clients forward traffic. The forward and return traffic forms a single bi-directional connection and shares Service Function Paths with similar set of Service Functions.

In the figure below, the Service Classifier handling traffic from Host B must be able to identify return traffic (flow 2) and select the Service Function Path with "DDoS". Flow 1 and 2 form a connection and traverse DDoS in both directions.



(a) Flows from Host A



(b) Flows from Host B

Figure 1: Forward and return flows between two hosts

4.1.2. Service Classifier to distinguish initiator and responder

Even if a Security Service Function requires receiving bi-directional traffic of a connection, it should not necessarily receive traffic initiated from all network segments for performance, availability, and scalability reasons. For example, a DDoS mitigator is configured to receive bi-directional traffic initiated from the Internet, but not for traffic initiated from the internal network.

Traffic initiated from a network segment should be classified independently. In Figure 1(b), the Service Classifier for Host B must identify traffic initiated by Host B (flow 3) and classify it

independently. Such traffic bypasses the DDoS Service Function in this example.

The Service Classifier must distinguish between flow 2 and flow 3, both of which are from Host B to Host A. In other words, it must be able to identify the initiator and responder of a connection.

A Service Classifier that keeps certain state would be able to handle the above requirements. The state should be accessible by each Service Classifier if there are multiple instances handling traffic sources from various network segments.

4.1.3. Service Classification based on network and application criteria

The Service Classifier evaluates SFC Policies (i.e. Service Policies) in order to determine the traffic and associated Service Function Paths. In the case of Security Service Functions, the Service Policies can contain match criteria derived from all OSI layers of the packet.

SFC classification is often based on network data, including but not limited to: Network interface port, VLAN, source and destination IP addresses, source and destination TCP and UDP ports, IP protocol, etc. These properties can be derived from the packet headers and are consistent across every packet of a flow.

There are match criteria that are desired by Security Service Functions that are either not present in the first packet, or are not present in every packet.

Those criteria may comprise "application data" from above the network layer, referred to as "application criteria". For example, a policy rule may state:

```
for all TLS traffic, run the traffic through Service Function "TLS Proxy"
```

Another example of an application layer policy rule is:

```
for all HTTP traffic with content containing file types of interest, run the traffic through Service Function "File Stream Scanner"
```

The Service Classifier for Security Service Functions needs to handle complex Service Policy. In some cases, this can be achieved by embedding the Service Classifier function into a Security Service Function, such that it can evaluate the application data as it becomes available.

4.1.4. Switching Service Function Paths based on inspection and scanning results

Network data is likely to be available on the first packet of the flow. When only network data is used as Service Policy match criteria, a stateful Service Classifier will be able to determine the forward and reverse Service Function Paths from the first packet (initial classification). The forward and reverse Service Function Paths remain unchanged for the entire life of the flow for these types of policies.

When the Service Policy contains application criteria, the policy rule may not be fully evaluated until several packets have passed through the chain. For example, TLS traffic can be identified only after the TLS Client Hello handshake message is observed.

Multiple classifiers may be required to provide sufficient classification granularity and complete a full evaluation of the Service Policy. In many cases, classification will be co-located with a Security Service Function that has the ability to inspect and scan the application data.

A new Service Function Path may be selected by a non-initial classification, different from the one determined by the initial classification.

The selection of a new Service Function Path can be reflected in the NSH Service Path Header as a new Service Path ID for the Service Function Forwarder to direct the packet accordingly.

The decision of a new Service Function Path often needs to be stored in the Service Classifier to ensure that subsequent packets of the flow follow the new path. This is because the data that triggers a new Service Function Path may be available from one particular packet only. For example, the packet with the TLS Client Hello message is used to identify a TLS session. Subsequent packets may not contain information for identifying the TLS sessions. All subsequent packets, without being classified again, must travel through the path with the "TLS Proxy" Service Function.

The Service Function that is new in the packet path (as part of the new Service Function Path) has to be able to handle not seeing earlier packets in the flow. Refer to Section 4.2 for more discussions on Service Functions.

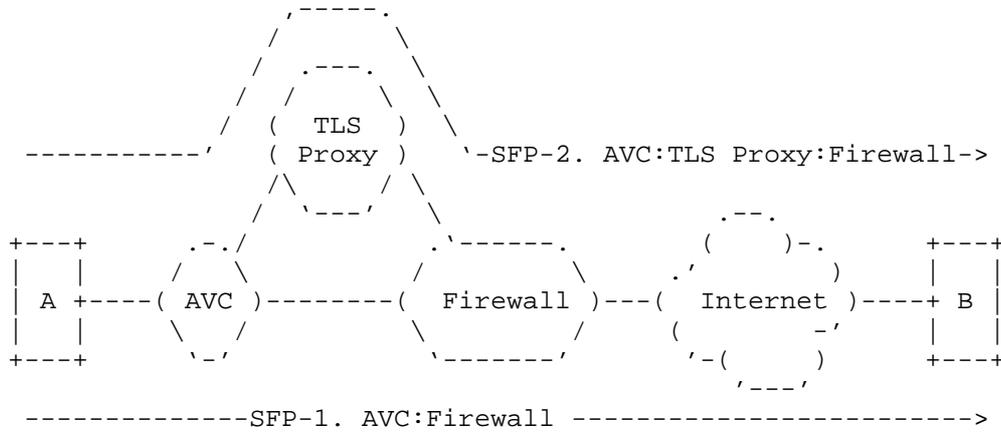


Figure 2: Mid-stream service function path update

Figure 2 illustrates a simple set of Security Functions deployed at the Internet edge. The default Service Function Path is SFP-1, with Service Functions "AVC" and "Firewall". When a TLS session is detected (e.g. by detecting the TLS Client Hello in the AVC Service Function), packets of the flow from that point on are switched to SFP-2, which contains "TLS Proxy" between "AVC" and "Firewall" to decrypt the TLS traffic for inspection.

Packets	Service Function Path
TCP Handshake	SFP-1. AVC:Firewall
TLS Client Hello	SFP-1; Switched to SFP-2 after AVC
Rest of TLS HS	SFP-2. AVC:TLS Proxy:Firewall
HTTPS Data	SFP-2. AVC:TLS Proxy:Firewall

Table 1: SFP taken by each packet in an HTTPS connection

Table 1 lists the Service Function Path for each packet in an HTTPS connection, from the TCP 3-way handshake to the HTTPS data packets. A new Service Function Path is selected in the middle of the connection after the TLS Client Hello is observed.

4.2. Service Function Use Cases

4.2.1. Service Classifier-capable Service Function

Service Functions that are capable of selecting a new Service Function Path must have the Service Classifier function integrated. Such Service Functions are often responsible for classification using their inspection and scanning results and updating Service Function Paths based on the Service Policy.

4.2.2. Service Functions operating on L5 or L7 data

Certain Security Service Functions operate on L5 to L7 data. For example, a "TLS Proxy" consumes a TCP stream without retransmitted or overlapping TCP segments. A "Web Proxy" operates on TCP stream of HTTP traffic. The data consumed by such Service Functions may not be in the original packet frame format, and the data may not contain the original L2-L4 header information. Such Service Functions can obtain the session or flow information from the SFC metadata carried in NSH.

4.2.3. Service Function mid-stream pick-up

When a new Service Function Path is selected as a result of Service Policy re-evaluation with application layer policy metadata, a new Service Function may need to start handling packet frames in the middle of a flow. This is referred to as "mid-stream pick-up". Although this is mid-stream from a flow perspective, it is still a complete data stream from the Service Function perspective (e.g., although "TLS Proxy" Service Function may not see the prior TCP handshake packets, it still sees the entire TLS stream). Similarly, transaction based Service Functions only handle packets belonging to a particular transaction. Such Service Function may use the flow ID metadata carried in NSH to link the session back to the flow.

Packet	AVC	TLS Proxy	Firewall
TCP SYN	X		X
TCP SYN/ACK	X		X
TCP ACK	X		X
TLS Client Hello	X	X	X
Rest of TLS HS	X	X	X
HTTPS Data	X	X	X

Table 2: Service Functions visited by each packet in an HTTPS connection

Table 2 lists the Service Functions visited by each packet from an HTTPS connection. The first packet that the Service Function "TLS Proxy" receives is the TLS Client Hello, as opposed to the TCP handshake packets prior to it.

4.2.4. Bypassing a particular Service Function

Certain Security Service Functions can be compute-intensive while only serving a particular task. It may be required to bypass such a Service Function in the middle of a flow. For example:

- o "Firewall" may request offloading of certain flows to fast forwarding engine with minimal inspection
- o "HTTP Inspector" may decide to not inspect video streams from a site with a high reputation
- o "TLS Proxy" may have to avoid decryption of banking traffic for compliance reasons

The decision to bypass a Service Function is made by the Service Function with its static policy, the inspection results and/or mid-stream evaluation of Service Policy.

Even if a flow is offloaded or bypassed, the Security Service Function may want to continue receiving critical packets for state tracking purposes. For example, "Firewall" may want to receive TCP control packets, and "HTTP Inspector" may want to track each transaction in the same flow.

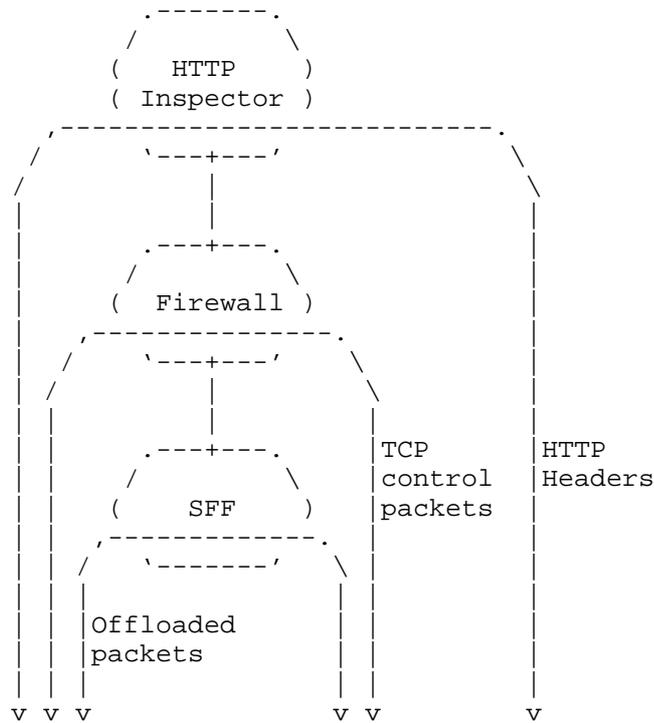


Figure 3: Service function bypass examples

A new Service Function Path may be selected to steer traffic to the path with the bypassed Service Function removed. The Service Function may update the NSH Service Path ID in the packet (in-band signaling) if the Service Function has knowledge of the relevant Service Function Paths. Alternatively, the Service Function may signal the Service Classifier (out-of-band) to update the Service Function Path for excluding the Service Function.

Service Function bypass may also follow the procedure described in "Service Function Simple Offloads" [I-D.kumar-sfc-offloads], where the Service Function signals the Service Function Forwarder to offload a flow, without selecting a new Service Function Path. The Service Function Forwarder caches the offload request and bypasses the Service Function in the service path for the remainder of the flow.

4.2.5. Receive-only Service Functions

Certain Service Functions such as an IDS may operate in "receive-only" mode, i.e. they consume a packet instead of passing the packet through. The Service Function Forwarder should send copies of packets to receive-only Service Functions.

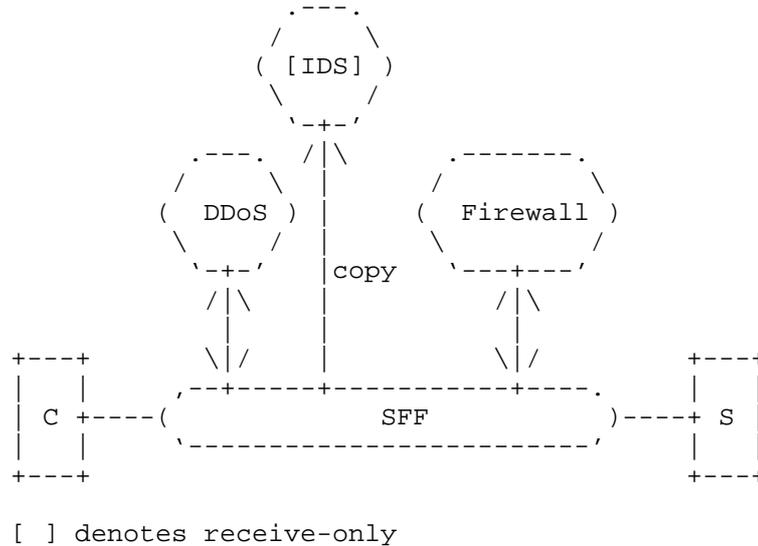


Figure 4: Receive-only service functions in SFC

Figure 4 illustrates an example of receive-only Service Function and its insertion into a Service Function Chain. The IDS Service Function receives copies of packets from the Service Function Forwarder.

4.3. Service Data Handling Use Cases

4.3.1. Service Function injected new packet

Security Service Functions may inject new packets into an existing flow in either direction. For example,

- o "Web Proxy" inserts an HTTP page challenging the client to login, in order to obtain the client's identity. This is in response to a packet (likely HTTP Request) but in the opposite direction of the flow.
- o "Firewall" checks an idle TCP connection by sending TCP keepalives to the client and/or server (known as "TCP dead connection

detection"). This is on existing flows but not responding to a prior packet.

- o "Firewall" sends ICMP error message after dropping a packet. This is in response to the prior packet but on a new flow.

The Service Function or Service Classifier needs to conduct a lookup of the reverse Service Function Path and populate the NSH Service Path Header. The approaches described in [I-D.penno-sfc-packet] may be adopted to support this use case.

4.3.2. Service Function initiated connections

A Service Function may need to create its own connections that are not associated with any client connection. Use cases include probing of servers behind a web proxy. In such cases, there will be no existing metadata for the Service Function to use to establish this connection. Such connections should be classified just like any other connections traversing the Service Function Path, as there may be Service Functions that are required to perform operations such as NAT on such connections in order for it to reach its destination.

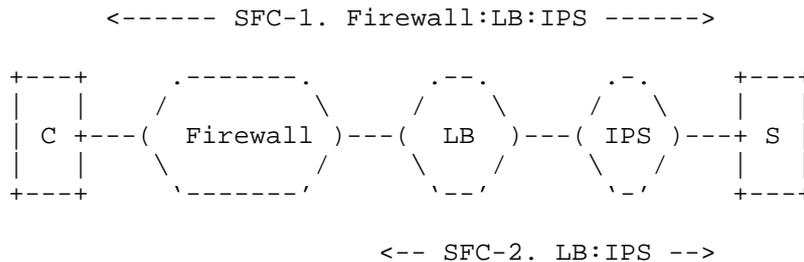


Figure 5: SFC for service function initiated connection

Option 1: Service Classifier in Service Function. A Service Classifier-capable Service Function may conduct service classification to determine the Service Function Path for the Service Function initiated connection. It can add an NSH with the proper Service Path Headers to the packets, and the Service Function would be the first SF on the chain. Response traffic follows a reverse Service Function Path and terminates at the Service Function. The number of Service Path Identifiers increases with more Service Functions bearing such capability.

Option 2: Service Classifier external to Service Function. A Service Function may send native packets without NSH when it is not capable of service classification. Such traffic is handled by the Service Classifier, which will populate the traffic with the appropriate NSH.

4.3.3. Security classification results

Security Service Functions may generate security classification results (e.g. policy actions and inspection results) while processing the packet data. Certain actions such as packet drop and flow closure can be taken immediately.

However, Service Functions can choose not to take any action immediately. Instead, it may pass the classification results to the subsequent Service Functions or to a control point.

Security classification results may be carried in NSH metadata as a score value. The score can be relayed and refined by other Security Service Functions along the path. Figure 6 below depicts an example of accumulating the client's score based on the Service Function's classification result. The client's reputation score is 6 as reported by the Service Function "Reputation", and the score is then passed to the next Service Function "Web Proxy" as the initial score for the connection. "Web Proxy" reduces the score to 3 after detecting access to a low reputation website. The Service Function "File Scanner" is involved due to the low score so far. After the "File Scanner" conducts scanning on the downloaded file and identifies it to be a malware, it updates the score to be -5 which is below the threshold for the connection to be blocked.

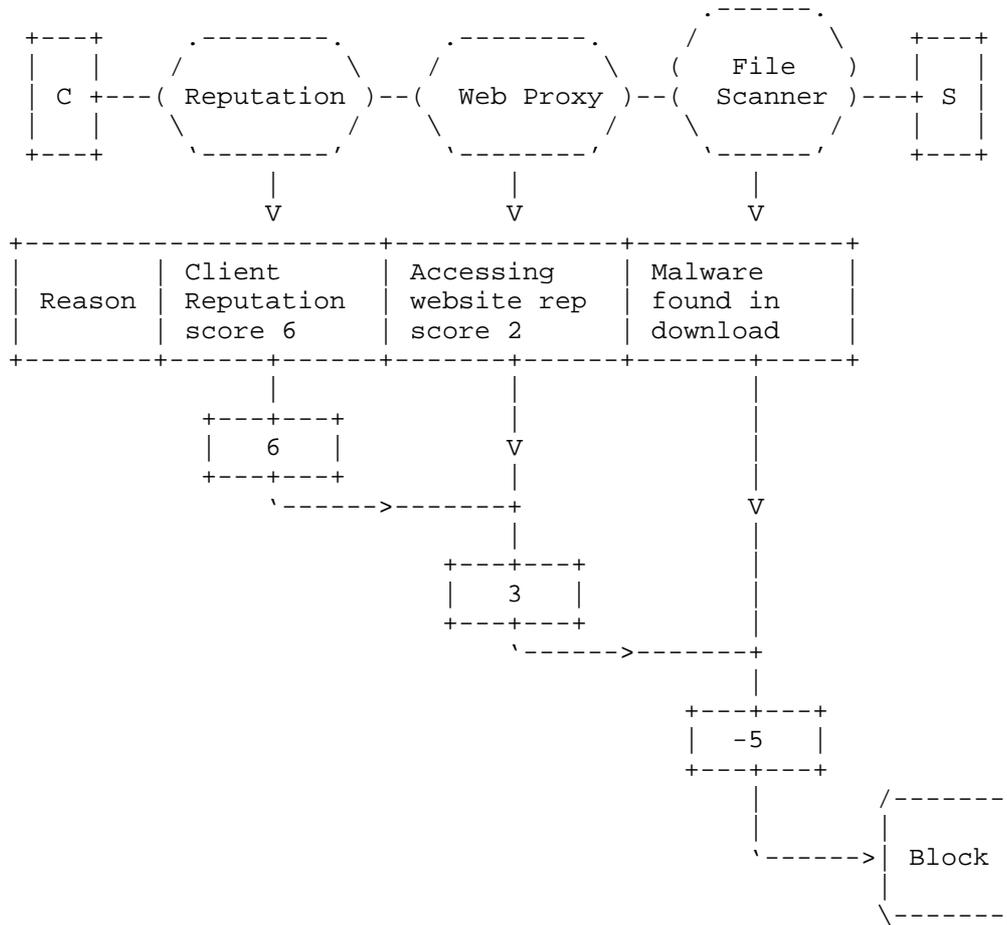


Figure 6: Security classification result with accumulated client score

Alternatively, each participating Service Function may send its own classification result to a central Service Function or control point for aggregation. Actions are then taken by a specific Service Function or control point based on the accumulated results. Figure 7 illustrates this option.

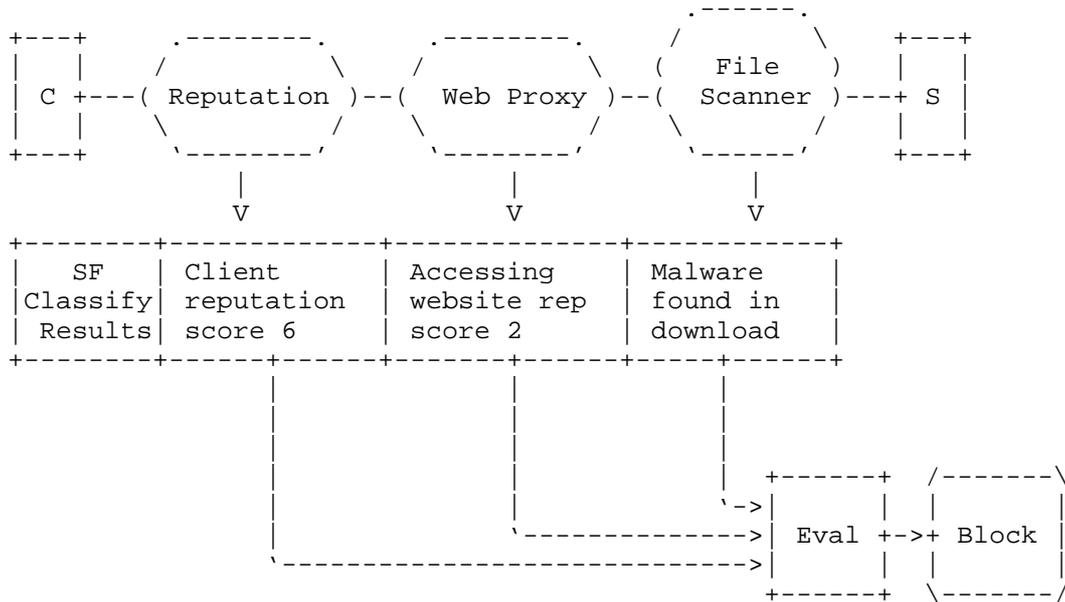


Figure 7: Aggregation of security classification results

5. General Requirements

The above use cases lead to the following requirements for applying SFC to security services on the data traffic.

Requirements that may need working group drafts:

1. SFC SHOULD allow packet frames carrying only L5 and upper layer traffic data without L2-L4 headers.
2. SFC SHOULD support bypass of a Service Function in the middle of a connection while allowing necessary control packets to reach the Service Function. Possible extension to [I-D.kumar-sfc-offloads]
3. SFC control plane and packet plane MUST support receive-only Service Functions.
4. SFC MUST support packet injection to the opposite direction of a Service Function Path. Possible extension to [I-D.penno-sfc-packet]
5. SFC SHOULD allow metadata passing classification results.

Requirements for implementation driven by respective use cases:

1. SFC MUST support the use of stateful Service Classifiers and Service Functions if present.
 2. Service Classifiers MUST have the ability to classify forward and the corresponding reverse Service Function Paths.
 3. Service Classifiers MUST be able to distinguish between traffic initiator and responder.
 4. SFC MUST support the use of Service Policies with network and application layer match criteria if supported by Service Classifier.
 5. SFC MUST support Service Function Path update or selection of a new path by a Service Classifier in the middle of a flow.
6. Security Considerations

This document describes use cases for Security Service Functions to participate in SFC. There are cases such as picking up traffic from the middle of a packet stream or handling packets without L2-L4 headers. Security Service Functions must process those types of traffic properly and associate them with the appropriate internal state.

While each Security Service Function applies its own implementation to secure the internal data, communications between Service Functions need to be secured as well. Measures must be taken to ensure metadata such as security classifications carried in NSH is not tampered.

7. Acknowledgments

The authors would like to thank Paul Quinn, Reinaldo Penno and Jim Guichard for their detailed review, comments and contributions.

8. IANA Considerations

This document includes no request to IANA.

9. References

9.1. Normative References

- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

9.2. Informative References

- [I-D.kumar-sfc-offloads]
Surendra, S., Guichard, J., Quinn, P., Halpern, J., and S. Majee, "Service Function Simple Offloads", draft-kumar-sfc-offloads-03 (work in progress), October 2016.
- [I-D.penno-sfc-packet]
Penno, R., Pignataro, C., Yen, C., Wang, E., Leung, K., and D. Dolson, "Packet Generation in Service Function Chains", draft-penno-sfc-packet-03 (work in progress), April 2016.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <<http://www.rfc-editor.org/info/rfc3022>>.

Authors' Addresses

Eric Wang
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: ejwang@cisco.com

Kent Leung
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: kleung@cisco.com

Jeremy Felix
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: jefelix@cisco.com

Jay Iyer
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: jiyer@cisco.com

Service Function Chaining
Internet-Draft
Intended status: Informational
Expires: April 28, 2017

E. Wang
K. Leung
Cisco Systems Inc.
October 25, 2016

Receive-Only Service Function and External Service in SFC
draft-wang-sfc-receive-only-01

Abstract

A category of services such as Intrusion Detection Service and Packet Capture operates in "receive-only" mode. They are "packet sinks" which consume all packets sent to them. This document describes the proposals for such service to be part of the Service Function Chaining framework.

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 April 23, 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 3
 2. Definition Of Terms 3
 3. Receive-Only Service Function vs. External Service 3
 4. SFC Packet Plane 6
 4.1. Receive-Only Service Function 6
 4.2. Receive-Only External Service 9
 5. SFC Control Plane 12
 5.1. Receive-Only Service Function 12
 5.2. Receive-Only External Service 12
 6. SFC Element Considerations 13
 6.1. Receive-Only Service Function 13
 6.2. Extended SFC Proxy 14
 6.3. Receive-Only External Service 14
 6.4. Service Function Forwarder 14
 6.4.1. Receive-Only Service Function 14
 6.4.2. Receive-Only External Service 14
 6.4.3. SFF Capabilities Considerations 15
 7. Security Considerations 16
 8. Acknowledgments 16
 9. IANA Considerations 16
 10. References 16
 10.1. Normative References 16
 10.2. Informative References 17
 Authors' Addresses 17

1. Introduction

Services in Service Function Chaining (SFC) usually operate in "inline" mode, where they process the received packets and return the packets to the Service Function Forwarder (SFF). Some services especially in the network security domain can buffer, inject or block certain packets, as well as proxy entire connections ([I-D.wang-sfc-ns-use-cases]). However, in general they still forward packets.

[I-D.wang-sfc-ns-use-cases] also describes a special set of services that consume all packets sent to them. We refer to such behavior as "receive-only". A receive-only service could be a Service Function (SF) participating in SFC ([RFC7665]), or it could be an External Service (ES) receiving packets from the SFF or another regular SF. This document describes proposals for designing SFC packet plane and control plane to incorporate receive-only services.

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 RFC 2119 [RFC2119].

2. Definition Of Terms

This document uses the terms as defined in [RFC7498], [RFC7665] and [I-D.ietf-sfc-nsh].

In addition the following terms are defined.

Receive-Only (RO): A service operational mode where the service does not forward on packets that it receives. It is often known as "packet sink" as well.

RO SF: A Receive-Only Service Function participating in SFC as defined in [RFC7665], except that the SF operates in Receive-Only mode.

RO ES: A Receive-Only External Service not participating in SFC. Specifically, an RO ES is not an SF. It is not allocated with a Service Index (SI) in the Service Function Path (SFP). However, it receives packets from the SFF or an SF.

Extended SFC Proxy: An SFC Proxy ([RFC7665]) with extended capability to support Receive-Only SF. The SFC Proxy replicates and sends packets to the RO SF. The SFC encapsulation may be preserved if the RO SF is SFC aware. The Proxy forwards the original packets back to the SFF in the same way as a regular SF (e.g., with Service Index (SI) decremented in NSH).

Flow: A unidirectional traffic stream identified by network layer attributes, specifically IP addresses and TCP/UDP ports for TCP/UDP traffic.

Connection: A bidirectional traffic stream composed of two flows sharing the same network layer attributes.

3. Receive-Only Service Function vs. External Service

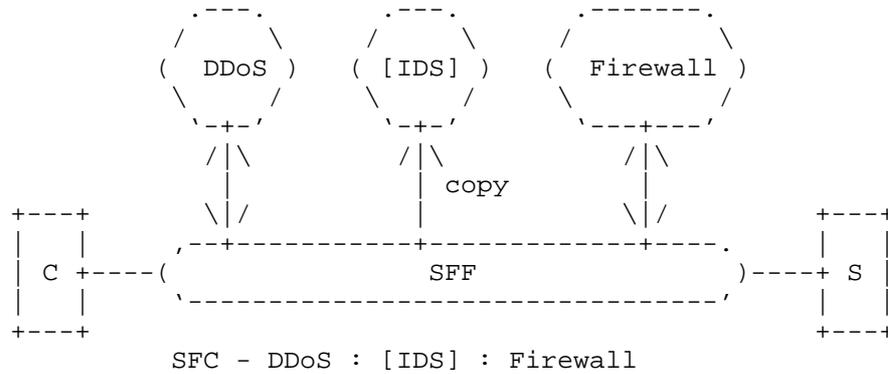
A receive-only service may be a Service Function (SF) or External Service (ES) depending on whether the SFC administrator designs the service to be part of the SFC or not.

In the following example, the IDS service is located between "DDoS Mitigator" and "Firewall" as part of an integrated security solution

to protect the server (S). Packets from the client (C) must be examined by a chain of services before they reach the server. The three service functions, DDoS, IDS and Firewall, compose an SFC. Each SF including the IDS SF is allocated with a Service Index (SI) in the SFP. The IDS SF is a fundamental part of the service chain with the only exception that it does not egress packets. We refer to the IDS service as a Receive-Only SF (RO SF).

There are other attributes for an RO SF. There is a limited number of location options for "IDS" to be deployed in an SFC. Once deployed, the location of "IDS" does not change usually unless the SFs are added to or removed from the SFC.

Extending the SFC framework to RO SF enables a common SFC policy model for SFC administrators. The administrator only needs to manage one set of policy that covers both RO and regular SFs, no matter how they handle packets.



[] denotes receive-only

Figure 1: Receive-Only Service Function (RO SF) example

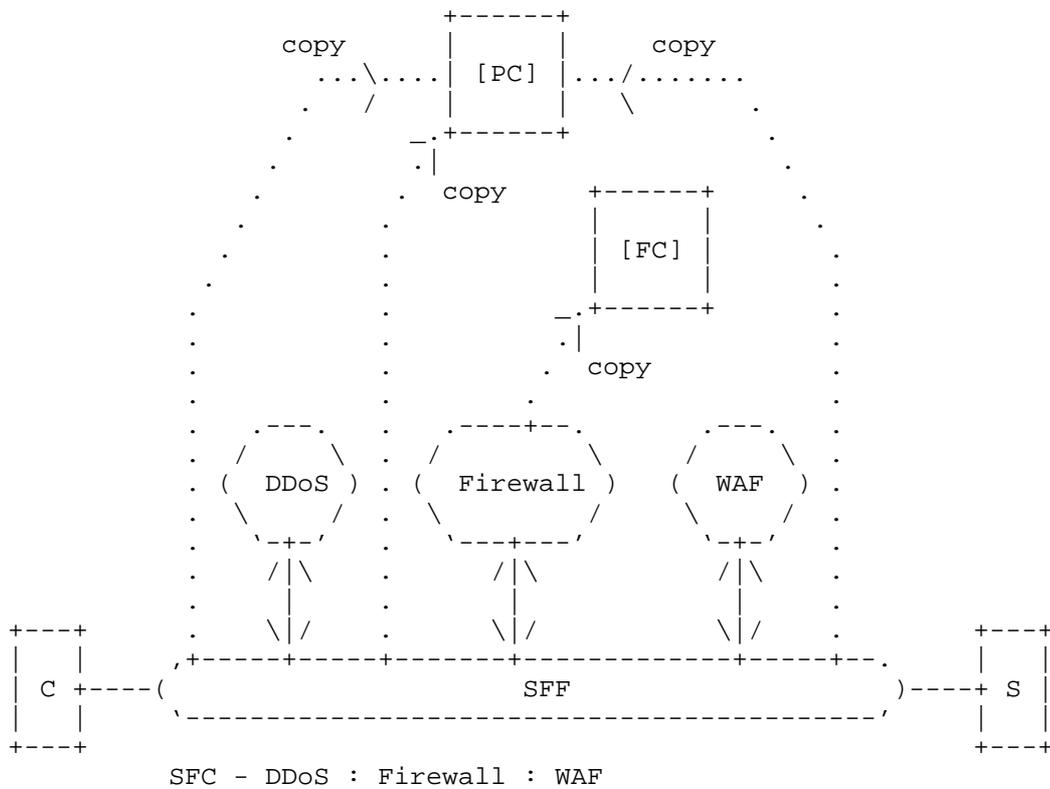
"Packet Capture" for troubleshooting represents the other type of receive-only services that do not participate in SFC.

The administrator may insert "Packet Capture" at any stage of an SFP. Packets may be captured before and/or after one or multiple SFs, which means there are many possible invocation points for "Packet Capture" in an SFP. The administrator may want to dynamically insert or remove "Packet Capture" based on the need for troubleshooting and threat analysis. When doing so, it is desired that the construction of the SFP is not affected. That is, the Service Path ID (SPID) and Service Index (SI) for each of the SFs in the SFP remain unchanged.

Services with the above attributes receive SFC packets but are not listed in an SFC. They do not consume an SI in the SFP. We refer to such a service as a Receive-Only External Service (RO ES).

An RO ES still requires support from SFC elements including SFF and SF for receiving packets. Because an RO ES does not send back packets, it must receive replication of the packets. Depending on the use cases and performance requirements, an SFF or SF may perform the packet replication for the RO ES. For example, a Packet Capture for troubleshooting purpose may tap on the SFF at one or more locations along the SFP (Figure 2).

This document describes the necessary enhancements to the SFC framework for supporting RO ES.



[] denotes receive-only

PC: Packet Capture

FC: File Capture

Figure 2: Receive-Only External Service (RO ES) examples

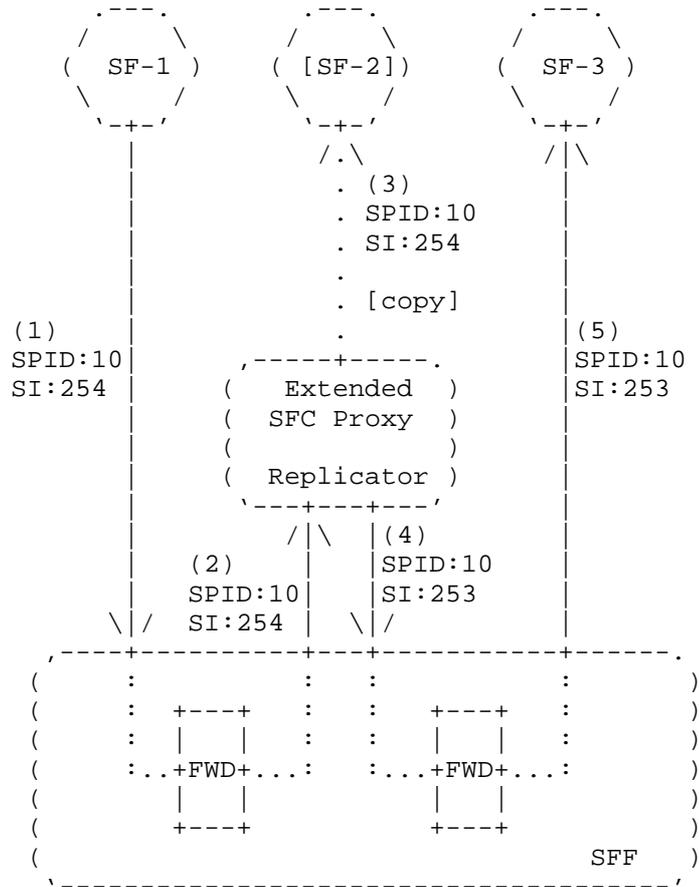
4. SFC Packet Plane

4.1. Receive-Only Service Function

A Receive-Only SF behaves the same way as a regular SF except that it does not forward packets. As a result, it must receive a copy of the packets while the original packets travel through the rest of the SFs in the SFP. Because an RO SF occupies an SI in the SFP, the SI of the original packet must be decremented after the packet passes the RO SF.

Considering the fact that an RO SF does not forward packets and SFF is not designed to decrement SI, an Extended SFC Proxy is leveraged to perform the packet replication and SI decrement tasks on behalf of the RO SF. As shown in the Figure below, the Extended SFC Proxy makes a copy of the packet including the NSH and sends the copy to the RO SF-2. It decrements the SI of original packet and forwards that packet back to the SFF. This capability is an extension to the current SFC Proxy as defined in ([RFC7665]).

From the SFF's perspective, the combination of the Extended SFC Proxy and the RO SF behaves as a regular SF that processes and forwards packets with SI decremented. From the RO SF's perspective, the Extended SFC Proxy may be a logical component in the specific SFF implementation for efficiency.



FWD: Forwarding Table Lookup

Figure 3: Packet Replication and SI Decrement by Extended SFC Proxy for RO SF

A packet goes through the following steps in the above example:

1. SFF receives the packet from SF-1 with SPID=10 and SI=254
2. SFF looks up in its forwarding table and finds Extended SFC Proxy for SF-2 to be the next hop. SFF sends the packet to SF-2's Proxy.
3. Extended SFC Proxy replicates the packet and sends the copy to SF-2 which is an RO SF. The copy has SI=254

4. Extended SFC Proxy decrements the SI of the original packet and sends the packet back to SFF. The packet now has SI=253
5. SFF looks up the next service function, SF-3, in its forwarding table based on the SPID and SI from Extended SFC Proxy. SFF sends the packet to SF-3.

4.2. Receive-Only External Service

A Receive-Only External Service still receives a copy of the packets designated to it even if it is not listed in the SFP.

For some use cases such as capturing a file content for sandbox analysis, packet data replication may be conducted by an SF capable of identifying file boundary in the packet stream. The RO ES would be associated with the SF and receives packet data from the SF directly.

Alternatively, for use cases such as generic packet capture for troubleshooting, the SFF may carry out the packet replication and forwarding work. Higher performance may be achieved with hardware based SFF.

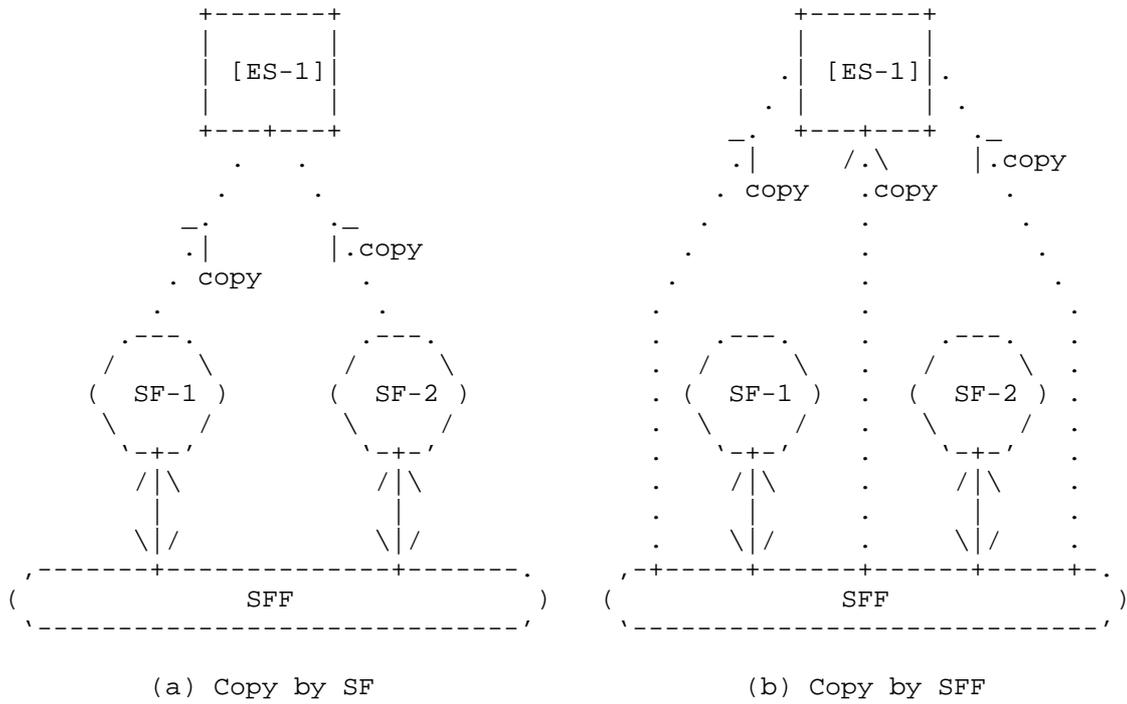


Figure 4: Packet replication options for RO ES

When an RO ES receives packets from the SFF, it may be attached to the SFF at multiple locations along the SFP. The location may be indicated by a (SPID, SI) pair and programmed into the SFF's forwarding table.

The SFF performs packet replication when the packet needs to be sent to the RO ES (SFC Proxy cannot be used because RO ES is not an SF). The SI of the original packet MUST NOT be affected by the fact that a copy is being sent to the RO ES. The following figure illustrates an example with SFF performing packet replication.

5. SFC Control Plane

5.1. Receive-Only Service Function

An RO SF such as IDS is specified the same way as a regular SF in the SFC Classification Policy. For example, the SFC in Figure 1 contains the following SFs:

```
SFC - DDoS : [IDS] : Firewall
```

When the SFC is converted to an SFP, the combination of Extended SFC Proxy and the IDS RO SF presents as a regular SF in the SFP as depicted in Figure 3. The SFP comprises of the following:

```
SFP - DDoS : SFC Proxy for [IDS] : Firewall
```

The SFC Control Plane also provisions the Extended SFC Proxy to send the replicated packet to the [IDS] SF. The provisioning message is outside the scope of this document.

5.2. Receive-Only External Service

An RO ES is not provisioned by SFC Classification Policy. Implementation may choose to use a dedicated policy for an RO ES such as "Packet Capture Policy".

The policy for RO ES may be configured into a regular SF when the SF performs replication for the RO ES.

In the scenario where SFF performs packet replication, the RO ES policy may be evaluated by the SFC Control Plane. The SFC Control Plane programs the replication locations into the SFF, as indicated by (SPID, SI) pairs. Figure 6 below illustrates a control plane flow for setting packet replication in the SFF for an RO ES.

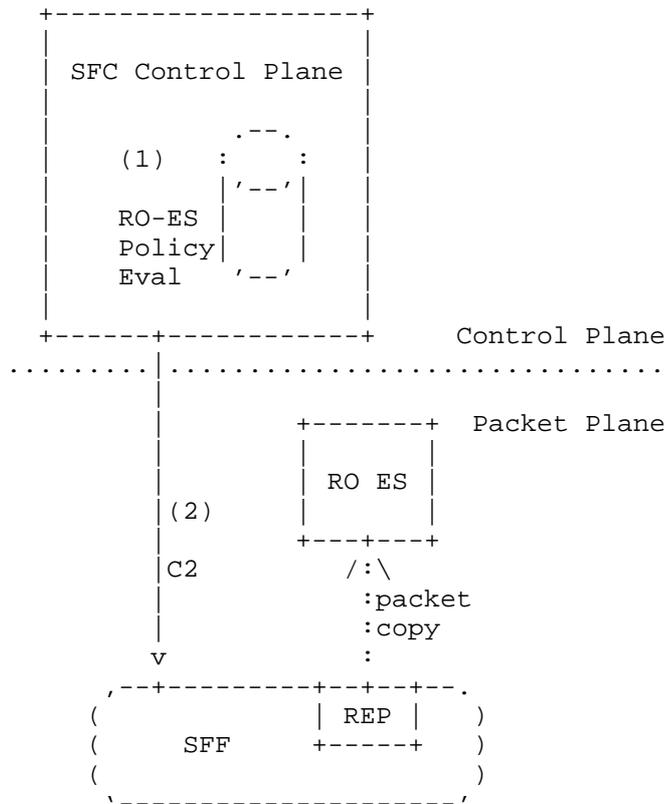


Figure 6: Control flow for SFF replication for RO ES

1. SFC Control Plane evaluates the RO ES policy and determines the location(s) for SFF to perform packet replication and send packets to RO ES.
2. SFC Control Plane uses C2 Control Plane-SFF interface ([I-D.ietf-sfc-control-plane]) to update the SFF forwarding table with replication entries to RO ES.

6. SFC Element Considerations

6.1. Receive-Only Service Function

An RO SF MUST NOT send received packets back to the Extended SFC Proxy.

6.2. Extended SFC Proxy

The Extended SFC Proxy for an RO SF carries the following additional capabilities compared with a regular SFC Proxy:

1. Packet replication
2. Preserving the SFC encapsulation in the copy of the packet when the RO SF is SFC aware

The Extended SFC Proxy MUST discard any packets from the RO SF to prevent duplicated packets to the SFF. When preserved, the NSH SPID/SI in the packet copy sent to RO SF MUST not change. The NSH SI in the original packet forwarded back to the SFF MUST be decremented.

6.3. Receive-Only External Service

An RO ES should have proper transport with the data source, either the SF or SFF. If it receives replicated packets from the SFF, the RO ES should comply with the transport as specified for the SFC, similar to that between a regular SF and the SFF.

An RO ES MUST NOT send received packets back to the data source.

6.4. Service Function Forwarder

6.4.1. Receive-Only Service Function

There is not any special requirement for the SFF to support an RO SF. The Extended SFC Proxy performs packet replication and other regular SF tasks on behalf of the RO SF.

6.4.2. Receive-Only External Service

The following figure illustrates a sample SFF forwarding table when the SFF carries out the packet replication task for RO ES. The "copy" column in the forwarding table is used to decide whether a copy or the original packet should be sent to the next hop (SF or ES). Entries for the RO ES have the "copy" field set.

SFF Forwarding Entries			
SPID	SI	Next Hop	Copy
...			
10	254	SF-1	
		ES-1	x
10	253	SF-2	
20	254	SF-1	
20	253	SF-3	
		ES-2	x
20	252	SF-4	
...			

Figure 7: Sample SFF forwarding table with RO ES entries

6.4.3. SFF Capabilities Considerations

In order to support RO ES, implementation of an SFF SHOULD support the following capabilities:

1. Packer replication
2. Discarding any packets returned by an RO ES

Additional capabilities for receive-only support include the following:

1. Replicating a portion of the packet (e.g. headers only)
2. Filtering selected packets to be replicated to receive-only service
3. Sending collective statistics in place of raw packet data
4. Producing Netflow/IPFIX and other events

Those capabilities are beyond the scope of this document.

7. Security Considerations

Even if an RO SF is required not to send packets back to the Extended SFC Proxy, the implementation of Extended SFC Proxy SHOULD handle packets from an RO SF gracefully without causing exceptions or duplicated packets in the SFP.

8. Acknowledgments

Authors would like to thank Jeremy Felix and Jay Iyer for their contributions, and Jim Guichard, Paul Quinn and Joel Halpern for their review and comments.

9. IANA Considerations

This document includes no request to IANA.

10. References

10.1. Normative References

- [I-D.ietf-sfc-control-plane]
Boucadair, M., "Service Function Chaining (SFC) Control Plane Components & Requirements", draft-ietf-sfc-control-plane-07 (work in progress), August 2016.
- [I-D.ietf-sfc-nsh]
Quinn, P. and U. Elzur, "Network Service Header", draft-ietf-sfc-nsh-10 (work in progress), September 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<http://www.rfc-editor.org/info/rfc2119>>.
- [RFC7498] Quinn, P., Ed. and T. Nadeau, Ed., "Problem Statement for Service Function Chaining", RFC 7498, DOI 10.17487/RFC7498, April 2015, <<http://www.rfc-editor.org/info/rfc7498>>.
- [RFC7665] Halpern, J., Ed. and C. Pignataro, Ed., "Service Function Chaining (SFC) Architecture", RFC 7665, DOI 10.17487/RFC7665, October 2015, <<http://www.rfc-editor.org/info/rfc7665>>.

10.2. Informative References

[I-D.wang-sfc-ns-use-cases]

Wang, E., Leung, K., Felix, J., and J. Iyer, "Service
Function Chaining Use Cases for Network Security", draft-
wang-sfc-ns-use-cases-01 (work in progress), March 2016.

Authors' Addresses

Eric Wang
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: ejwang@cisco.com

Kent Leung
Cisco Systems Inc.
170 W Tasman Dr
San Jose, CA 95134
U.S.A.

Email: kleung@cisco.com