International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

X.1045

(10/2019)

SERIES X: DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

Information and network security – Network security

Security service chain architecture for networks and applications

Recommendation ITU-T X.1045

International Telecommunication Union

ITU-T X-SERIES RECOMMENDATIONS

DATA NETWORKS, OPEN SYSTEM COMMUNICATIONS AND SECURITY

PUBLIC DATA NETWORKS	X.1–X.199
OPEN SYSTEMS INTERCONNECTION	X.200–X.299
INTERWORKING BETWEEN NETWORKS	X.300–X.399
MESSAGE HANDLING SYSTEMS	X.400–X.499
DIRECTORY	X.500–X.599
OSI NETWORKING AND SYSTEM ASPECTS	X.600–X.699
OSI MANAGEMENT	X.700–X.799
SECURITY	X.800–X.849
OSI APPLICATIONS	X.850–X.899
OPEN DISTRIBUTED PROCESSING	X.900–X.999
INFORMATION AND NETWORK SECURITY	
General security aspects	X.1000–X.1029
Network security	**X.1030–X.1049**
Security management	X.1050–X.1069
Telebiometrics	X.1080–X.1099
SECURE APPLICATIONS AND SERVICES (1)	
Multicast security	X.1100–X.1109
Home network security	X.1110–X.1119
Mobile security	X.1120–X.1139
Web security	X.1140–X.1149
Security protocols (1)	X.1150–X.1159
Peer-to-peer security	X.1160–X.1169
Networked ID security	X.1170–X.1179
IPTV security	X.1180–X.1199
CYBERSPACE SECURITY	
Cybersecurity	X.1200–X.1229
Countering spam	X.1230–X.1249
Identity management	X.1250–X.1279
SECURE APPLICATIONS AND SERVICES (2)	
Emergency communications	X.1300–X.1309
Ubiquitous sensor network security	X.1310–X.1319
Smart grid security	X.1330–X.1339
Certified mail	X.1340–X.1349
Internet of things (IoT) security	X.1360–X.1369
Intelligent transportation system (ITS) security	X.1370–X.1389
Distributed ledger technology security	X.1400–X.1429
Distributed ledger technology security	X.1430–X.1449
Security protocols (2)	X.1450–X.1459
CYBERSECURITY INFORMATION EXCHANGE	
Overview of cybersecurity	X.1500–X.1519
Vulnerability/state exchange	X.1520–X.1539
Event/incident/heuristics exchange	X.1540–X.1549
Exchange of policies	X.1550–X.1559
Heuristics and information request	X.1560–X.1569
Identification and discovery	X.1570–X.1579
Assured exchange	X.1580–X.1589
CLOUD COMPUTING SECURITY	
Overview of cloud computing security	X.1600–X.1601
Cloud computing security design	X.1602–X.1639
Cloud computing security best practices and guidelines	X.1640–X.1659
Cloud computing security implementation	X.1660–X.1679
Other cloud computing security	X.1680–X.1699
QUANTUM COMMUNICATION	X.1700–X.1729

For further details, please refer to the list of ITU-T Recommendations.

Recommendation ITU-T X.1045

Security service chain architecture for networks and applications

Summary

Recommendation ITU-T X.1045 supports provision of customized dynamic and adaptive security services for networks and applications. This Recommendation defines the security service chain and an architecture design for the security service chain. This Recommendation applies the security service chain to networks and applications. This Recommendation also enables tracing network attacks to their resources in a service function chain (SFC) overlay network with high performance and the mitigating/preventing of those attacks automatically.

History

Edition	Recommendation	Approval	Study Group	Unique ID*
1.0	ITU-T X.1045	2019-10-29	17	11.1002/1000/14043

Keywords

Interworking security service chain, network service header, security service chain, service function chain.

* To access the Recommendation, type the URL http://handle.itu.int/ in the address field of your web browser, followed by the Recommendation's unique ID. For example, http://handle.itu.int/11.1002/1000/11830-en.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications, information and communication technologies (ICTs). The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure, e.g., interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at http://www.itu.int/ITU-T/ipr/.

Table of Contents

Introduction

Network function virtualization (NFV) reduces capital expenditure and operational expenditure [b-FIS 2010] by replacing proprietary hardware-based network equipment with software-based virtualized network functions (VNFs) that can be instantiated dynamically on commodity servers (e.g., x86 based systems) and located more flexibly in the network. Software defined networking (SDN) [b-ONF SDN] enables automated and dynamic application deployment and reconfiguration due to its network-wide and fine-grained control of the network flows. These two complementary technologies (i.e., NFV and SDN) open up new ways to deploy network services and applications rapidly and automatically using service function chain (SFC) [IETF RFC 7665] in a more efficient and scalable fashion. Correspondingly, security as one of key features for network services and applications also has to be provided dynamically and adaptively to meet different security requirements for today's rapid-increasing and evolving networks and applications.

However, traditional security appliances such as Firewall or intrusion detection system (IDS) are implemented as hardware-based middleboxes and placed at fixed locations in the network. It is one of the administrators' duties to overload path selection mechanisms to force traffic through the desired sequences of security middleboxes as defined within security policies [b-SIGCOMM]. Since these security middleboxes are hardware-based and rather static, it is difficult to meet different security requirements for today's networks and applications. Moreover, it would be a heavy and complicated workload for the administrator to overload path selection mechanisms when traditional security appliances are deployed in a large-scale network. It would be even worse when those path selection mechanisms overloaded by the administrator are inconsistent.

This Recommendation aims to design security service chain architecture and provide customized security services dynamically and adaptively for networks and applications based on security service chain. It also supports tracing network attacks to their sources and mitigating/preventing those attacks automatically so that the administrator is not involved in responding to network attacks and security incidents.

Recommendation ITU-T X.1045

Security service chain architecture for networks and applications

1 Scope

This Recommendation discusses the security service chain in order to provide customized security services dynamically and adaptively for networks and applications. This Recommendation:

– defines the security service chain and an architecture for security service chain;

– designs procedures for security service chain creation; and

– applies security service chain to datacentres, intelligent transport systems (ITSs) and cyber defence in order to provide customized security services.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T X.1043] Recommendation ITU-T X.1043 (2019), *Security framework and requirements for service function chaining based on software-defined networking*.

[IETF RFC 7665] IETF RFC 7665 (2015), *Service Function Chaining (SFC) Architecture.*

[IETF RFC 8300] IETF RFC 8300 (2018), *Network Service Header (NSH)*.

3 Definitions

3.1 Terms defined elsewhere

This Recommendation uses the following terms defined elsewhere:

3.1.1 classification [IETF RFC 7665]: Locally instantiated matching of traffic flows against policy for subsequent application of the required set of network service functions. The policy may be customer/network/service specific.

3.1.2 classifier [IETF RFC 7665]: An element that performs classification.

3.1.3 service function chain (SFC) [IETF RFC 7665]: A service function chain defines an ordered set of abstract service functions and ordering constraints that must be applied to packets and/or frames and/or flows selected as a result of classification. An example of an abstract service function is "a firewall". The implied order may not be a linear progression as the architecture allows for SFCs that copy to more than one branch, and also allows for cases where there is flexibility in the order in which service functions need to be applied. The term "service chain" is often used as shorthand for service function chain.

3.1.4 service function (SF) [IETF RFC 7665]: A function that is responsible for specific treatment of received packets. A service function can act at various layers of a protocol stack (e.g., at the network layer or other OSI layers). As a logical component, a service function can be realized as a virtual element or be embedded in a physical network element. One or more Service Functions can be embedded in the same network element. Multiple occurrences of the service function can exist in the same administrative domain.

3.1.5 service function forwarder (SFF) [IETF RFC 7665]: A service function forwarder is responsible for forwarding traffic to one or more connected service functions according to information carried in the SFC encapsulation, as well as handling traffic coming back from the SF. Additionally, an SFF is responsible for delivering traffic to a classifier when needed and supported, transporting traffic to another SFF (in the same or different type of overlay), and terminating the service function path (SFP).

3.1.6 service function path (SFP) [IETF RFC 7665]: The service function path is a constrained specification of where packets assigned to a certain service function path must go. While it may be so constrained as to identify the exact locations, it can also be less specific. The SFP provides a level of indirection between the fully abstract notion of service chain as a sequence of abstract service functions to be delivered, and the fully specified notion of exactly which SFF/SFs the packet will visit when it actually traverses the network. By allowing the control components to specify this level of indirection, the operator may control the degree of SFF/SF selection authority that is delegated to the network.

3.1.7 SFC encapsulation [IETF RFC 7665]: The SFC encapsulation provides, at a minimum, SFP identification and is used by the SFC-aware functions, such as the SFF and SFC-aware SFs. The SFC encapsulation is not used for network packet forwarding. In addition to SFP identification, the SFC encapsulation carries metadata including data-plane context information.

3.1.8 SFC proxy [IETF RFC 7665]: Removes and inserts SFC encapsulation on behalf of an SFC-unaware service function. SFC proxies are logical elements.

3.1.9 network service header (NSH) [IETF RFC 8300]: Defines a new data-plane protocol, which is an encapsulation for SFCs. The NSH is designed to encapsulate an original packet or frame and, in turn, be encapsulated by an outer transport encapsulation (which is used to deliver the NSH to NSH-aware network elements). The NSH is composed of the following elements: a) Service Function Path identification; b) indication of location within a Service Function Path; c) optional, per-packet metadata (fixed-length or variable).

3.1.10 NSH-aware [IETF RFC 8300]: NSH-aware means SFC-encapsulation-aware, where the NSH provides the SFC encapsulation.

3.2 Terms defined in this Recommendation

This Recommendation defines the following terms:

3.2.1 metadata: Provides the ability to exchange context information between classifiers and SFs, and among SFs. The metadata, or context information shared between Classifiers and SFs, and among SFs, is carried on the NSH's Context Headers. It allows summarizing a classification result in the packet itself, avoiding subsequent re-classifications. Examples of metadata include classification information used for policy enforcement and network context for forwarding after service delivery.

NOTE – Definition adapted from [IETF RFC 7665] and [IETF RFC 8300].

3.2.2 Security service chain (SSC): A type of service function chain (SFC), a security service chain defines an ordered set of security functions and ordering of security policies that must be applied to packets and/or flows selected as a result of classification.

4 Abbreviations and acronyms

This Recommendation uses the following abbreviations and acronyms:

ACL Access Control List

API Application Programming Interface

DSP Data Service Provider

GDPR	General Data Protection Regulation
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ITS	Intelligent Transport System
NSH	Network Service Header
RBAC	Role Based Access Control
RSU	Road Side Unit
SAAR	Security Analytics and Automatic Response
SDN	Software Defined Networking
SF	Service Function
SFC	Service Function Chain
SFF	Service Function Forwarder
SFP	Service Function Path
SSC	Security Service Chain
SV	Special Vehicle
VPN	Virtual Private Network

5 Conventions

In this Recommendation:

The keywords **"is required to"** indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords **"is recommended"** indicate a requirement which is recommended but which is not absolutely required. Thus, this requirement need not be present to claim conformance.

The keywords **"is prohibited from"** indicate a requirement which must be strictly followed and from which no deviation is permitted, if conformance to this Recommendation is to be claimed.

The keywords **"can optionally"** indicate an optional requirement which is permissible, without implying any sense of being recommended. This term is not intended to imply that the vendor's implementation must provide the option, and the feature can be optionally enabled by the network operator/service provider. Rather, it means the vendor may optionally provide the feature and still claim conformance with the specification.

Italics are used in this Recommendation to indicate functions.

6 Overview of security service chain

A security service chain (SSC), a type of service function chain (SFC) as defined in [IETF RFC 7665], defines an ordered set of security functions and ordering of security policies that must be applied to packets and/or flows selected as a result of classification. An SSC enables security service providers to manage and operate stand-alone security services as well as to provide customized security services which can be integrated into other services such as an intelligent transport system (ITS), video services and location services. In this way, service providers (e.g., ITS service providers, video service providers, location service providers, etc.) can focus mainly on their own basic service logics and get professional security services from security service providers in order to provide secure services to the users.

Based on SSC deployment, there are two kinds of security service chains shown in Figure 6-1. Figure 6-1(a) shows a stand-alone SSC, which enables security service providers to manage and operate stand-alone security services by themselves. Figure 6-1(b) shows the other kind of security service chain, an interworking SSC, which will be integrated into other service function chains to deliver end to end services/applications.

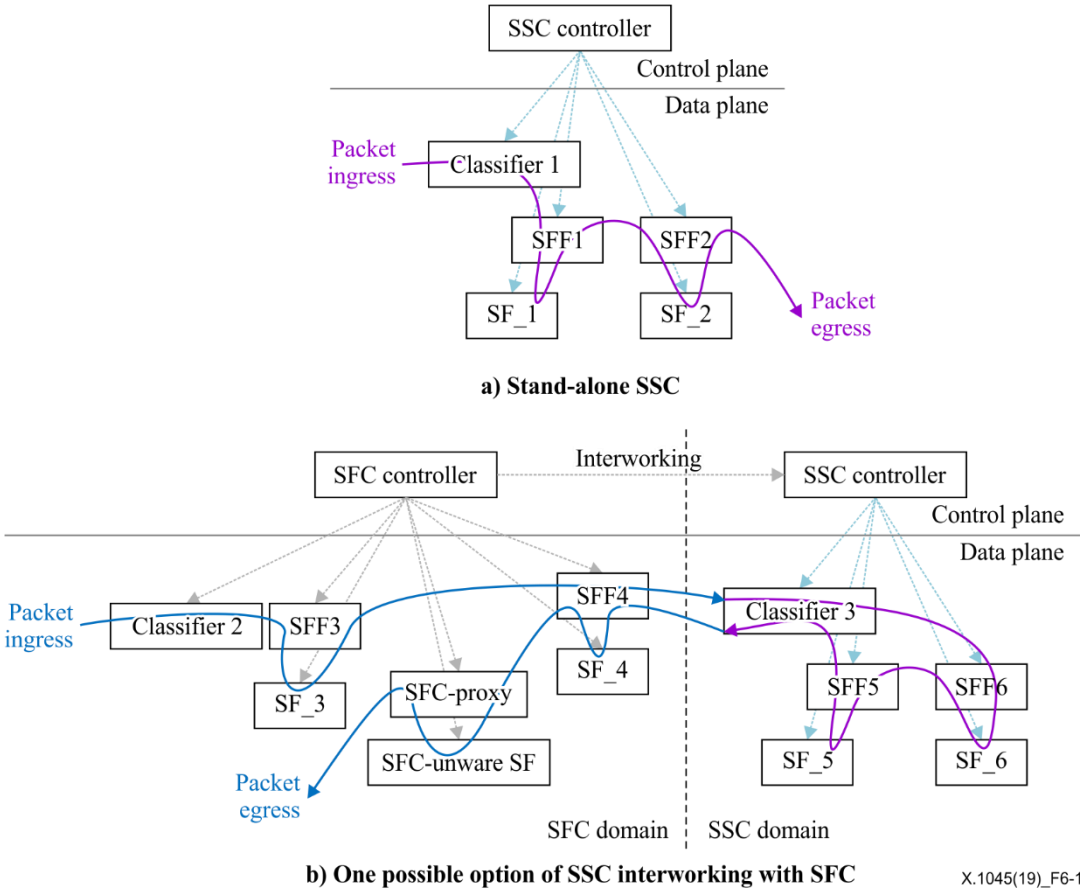


Figure 6-1 – Two kinds of security service chain

NOTE – The data flow path from the packet ingress to the packet egress in Figure 6-1(b) shows the data flow/packet is transported from an SFC to an SSC and to another SFC. There may be other data flow paths of SSC interworking with SFC, e.g., the data flow/packet can be transported starting from an SSC to an SFC and then to another SSC. The data flow path of an SSC interworking with an SFC needs to be designed according to the requirements of service/application.

Based on whether an SSC is related to a specific user or not, there are two types of security service chains. One type is the non-user-oriented SSC, which is neither specific to a user nor to a specific service logic. A non-user-oriented SSC provides security services such as packet filtering, intrusion detection and prevention and traffic cleaning through security appliances such as Firewall, intrusion detection system (IDS) and intrusion prevention system (IPS).

The other type of SSC is a user-oriented SSC, which is specific to a user and/or a specific service logic and will be integrated into a service function chain (SFC) with a specified order through cooperating with a SFC controller. For example, an end user has to be authenticated and authorized before starting to access services. A user-oriented SSC provides security services such as authentication, authorization and per-user encryption/decryption to achieve higher security level protection. A non-user-oriented SSC can be deployed as both a stand-alone SSC and an interworking SSC. A user-oriented SSC can be deployed only as an interworking SSC since it has to be integrated into a specific service logic.

7 Architecture of security service chain

7.1 Components of SSC architecture

Figure 7-1 shows the architecture of a security service chain which enables security service providers to create a stand-alone SSC to then manage and operate stand-alone security services such as packet filtering, intrusion detection and prevention and traffic cleaning.

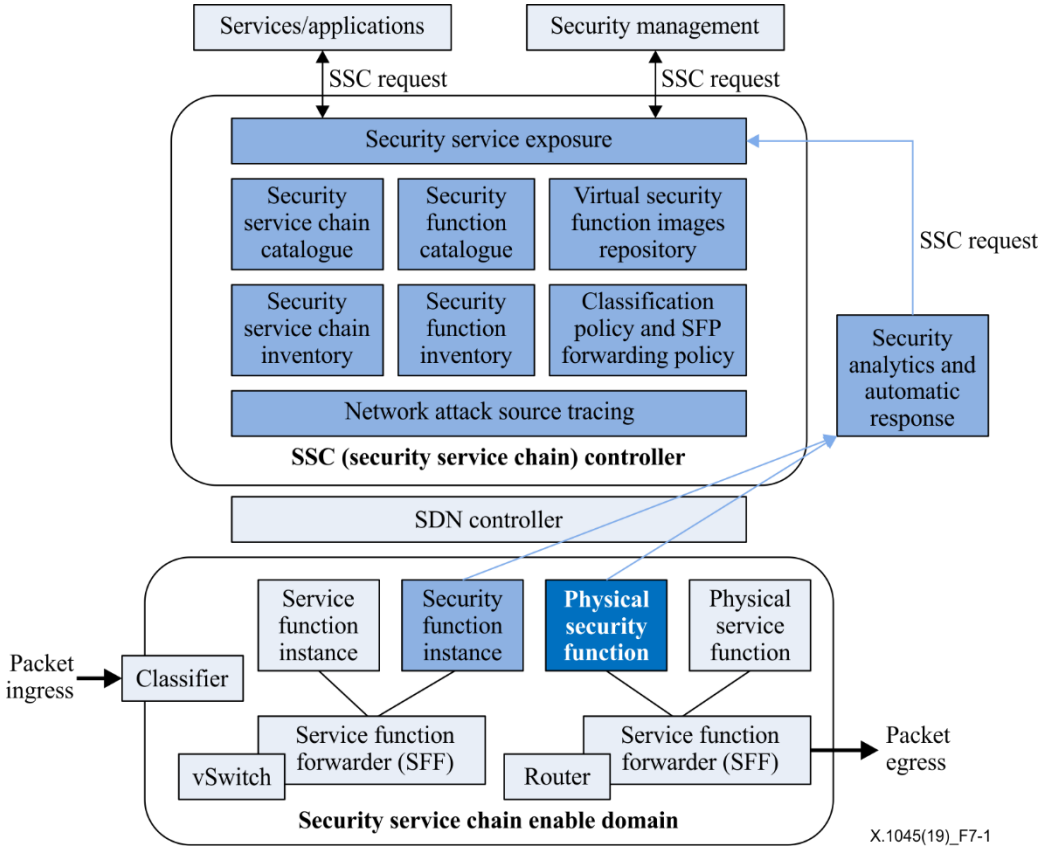


Figure 7-1 – Security service chain (SSC) architecture

In Figure 7-1, the *classifier*, *service function forwarders* (SFFs) and *service functions* (SFs) are as defined in [IETF RFC 7665]. Service functions include virtualized functions (e.g., *service function instance* and *security function instance*) and physical functions (e.g., *physical service function* and *physical security function*). The *SDN controller*, *vSwitch* and *router* are as defined in [b-ONF OpenFlow]. *Services/applications* and *security management* send an SSC request to the *SSC controller* according to their security requirements and then obtain customized security services from the *SSC controller*.

The functionalities of the SSC controller components and security analytics and automatic response (SAAR) are described in detail hereafter.

7.1.1 Security service exposure

Security service exposure is a logical function, which creates security service chains and exposes the corresponding security service chains' capabilities to *services/applications* through a set of application programming interfaces (APIs). According to Figure 7-1, after receiving a security service chain (SSC) request from *services/applications*, *security management* and *SAAR*, the *security service exposure* looks up the *security service chain catalogue* and selects an appropriate SSC (i.e., an ordered set of security functions) from the predefined SSC templates. Then, it checks if an instance of such a security service chain is registered and active in a *security service chain inventory*. If yes, this active instance will be selected and reused. Otherwise, it looks up the *security*

function catalogue and selects appropriate security functions according to the selected security service chain. Next, it checks if instances of these security functions are registered and active in a *security function inventory*. If yes, these active instances will be selected and reused for this security service chain. Otherwise, it looks up a *virtual security function images repository* and selects security functions images which will be instantiated and deployed in the appropriate hosts with the network topology provided by a cloud manager (e.g., Openstack) and software defined networking (SDN) controller. In this case, after instantiating the selected security functions images, these installed instances of security functions and this created instance of security service chain will be recorded in a *security function inventory* and in a *security service chain inventory* respectively. Finally, classification policy and service function path (SFP) forwarding policy are generated and inserted/updated into the policy tables of the *classifier* and *SFFs* respectively.

NOTE – An instanced virtual security function in a security service chain can be scaled according to its workload by a cloud manager.

7.1.2 Security service chain catalogue and security service chain inventory

The *security service chain catalogue* includes predefined templates of security service chains such as the identifier of the security service chain, the security service chain description, the list of selected security functions, the service function path (SFP) providing the ordered sequence of these security functions, the status of the instance in the *security service chain inventory*, how to find the information of each security function in the *security function catalogue*, etc.

The *security service chain inventory* includes the operational status of security service chain instances such as an identifier of the security service chain instance, list of instances of security functions, SFP of the security service chain instance, how to find the information of each security function instance in the *security function inventory*, etc.

7.1.3 Security function catalogue and security function inventory

Security functions include authentication, authorization, Firewall, IDS, DPI, traffic cleaning, etc. All these security functions are implemented as virtual functions or physical functions.

The *security function catalogue* includes static information of security functions such as an identifier of the security function, function description, requirements of deployment (e.g., CPU and memory), status of the instance in the *security function inventory*, how to find the information of security function images in the *virtual security function image repository*, etc.

The *security function inventory* includes operational status of security function instances such as an identifier of the security function instance, status of deployment (e.g., throughput and latency), etc.

7.1.4 Virtual security function images repository

A *virtual security function images repository* is a database to store images of virtualized security functions. When necessary, selected images should be transported with confidentiality and integrity protection to a remote site or host then instantiated in the security service chain enable domain.

7.1.5 Classification policy and SFP forwarding policy

In order to support creating a security service chain, the classification policy in the *classifier* and SFC control plane defined in [IETF RFC 7665] should be extended to reflect security related policies for binding an incoming flow/packet to a given SSC and SFP. The attributes/profiles of the classification policy may include a 5-tuple, a transport port or set of ports, part of the packet payload, data label of packet payload, a user identifier, a service identifier, a service type, a classification type, a classifier identifier, a SFP identifier, an owner of the classification policy (e.g., who generated the classification policy), the role of the classification policy generator, a next hop locator and one or more actions (e.g., forwarding, dropping, etc.).

In order to support creating security chain, the SFP forwarding policy table in the *SFF* and SFC control plane defined in [IETF RFC 7665] should be extended to reflect security related policies. The attributes/profiles of SFP forwarding policy may include a classifier identifier, a SFP identifier, a SFF identifier, an owner of the forwarding policy (e.g., who generated the forwarding policy), the role of forwarding policy generator, a next hop locator and one or more actions (e.g., forwarding, dropping, etc.).

To avoid policy conflicting when inserting/updating classification policy and SFP forwarding policy in corresponding policy tables, the *security service chain controller* should support a fine grained naming scheme for classification policy and a SFP forwarding policy [b-ICIN], which support authorization mechanisms such as access control list (ACL) and role based access control (RBAC).

7.1.6 Network attack source tracing

The *network attack source tracing* function is a logical function which supports tracing network attacks to their sources in the SFC overlay network with higher performance and lower tracing time comparing with traditional IP-based tracing back techniques in [b-IEEE IP Traceback] and [b-CMU Tracing].

When a network attack is detected, firstly the function *network attack source tracing* can trace the attack to the service function path (SFP) based on the flow/packet features (e.g., service path identifier (SPI), classifier identifier, SFF identifier); then notify the classifier and SFFs belonging to this SFP to do flow/packet matching at the same time. Those related classifiers and SFFs respond with their packet matching results. Finally, the attack source can be identified based on those packet matching results. In this way, the hops (classifier and SFFs belonging to a specified SFP) can be involved in traceback at the same time. So the time for tracing back is reduced and the performance is improved since traditional IP-based tracing back technique requires doing traceback hop-by-hop. The detailed procedures will be described in clause 9.2.

7.1.7 Security analytics and automatic response (SAAR)

The *security analytics and automatic response (SAAR)* function is a logical function which periodically fetches security events from security functions. Security functions may also actively report security events to the *SAAR* if they encounter security attacks. Moreover, the SAAR function also has the capability to contact the *network attack source tracing* function to enable tracing network attacks to their sources.

The *SAAR* analyses the above security events, finds appropriate security countermeasures, then sends an SSC request to the *security service chain controller*. After receiving the SSC request, the *security service chain controller* creates a corresponding security service chain and distributes classification policy and SFP forwarding policy to the *classifier* and *SFFs* together with security functions respectively. If the network attack sources (e.g., SFs, SFFs, classifier) are identified, new service function chains with corresponding SFPs will be created by replacing those attacked SFs/SFFs/classifier with other normal/unattacked ones. These attacks can then be prevented or mitigated. Therefore, tracing network attacks to their sources and preventing/mitigating these attacks automatically in the SFC overlay network can be implemented.

7.2 Supporting SSC interworking with SFC

The delivery of end-to-end services/applications often requires various service functions, which may be implemented in hosts, managed and operated by different administrative organizations or service providers. These hosts may locate in a geographically dispersed network. Moreover, security protection such as authentication and packet filtering for end-to-end services/applications is provided by a third trusted party (e.g., security service provider). Therefore, it is necessary to support SSC interworking with other SFCs.

Figure 7-2 shows how to extend SSC architecture, specified in clause 7.1, to support SSC interworking with other SFCs.

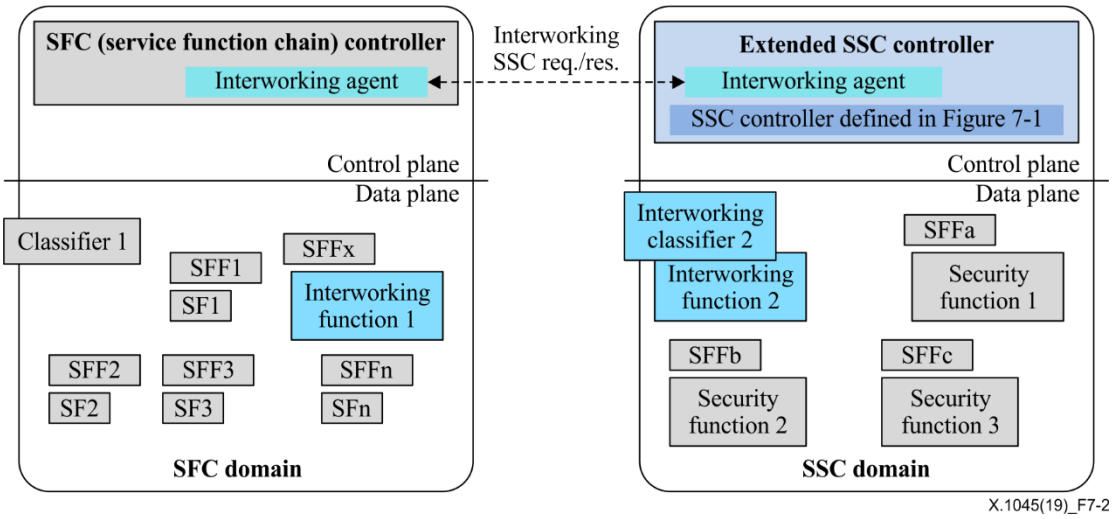


Figure 7-2 – SSC interworking with SFC

In Figure 7-2, some logic functions have the same definitions in [IETF RFC 7665] with no changes. These logic functions are the classifier (e.g., *Classifier1*), SFFs (e.g., *SFF1*, *SFF2*, *SFF3*, *SFFn*, *SFFx*, *SFFa*, *SFFb*, *SFFc*), SFs (e.g., *SF1*, *SF2*, *SF3*, *SFn*, *security function1*, *security function2*, *security function3*).

In order to support SSC interworking with SFC, there are two new logic functions compared with Figure 7-1, in light blue blocks (i.e., *interworking agent* and *interworking function*) and an extended classifier (i.e., *interworking classifier2*), which will be specified in this Recommendation.

The new logical function *interworking agent* should be implemented in an extended SSC controller and SFC controller and has the following capabilities:

- To judge if the delivery of end-to-end services/applications requires a service function chain interworking with a SFC in another domain.
- To manage the registration of SFC domains to be interworked with. For example, *Interworking Agent1* of the SFC controller manages the registration of an extended SSC controller. The registration information includes (but is not limited to) SFC domain identification, SFC domain name, IP address and port of SFC controller, functionality description of SFC domain, etc.

NOTE – SSC is a type of SFC. In order support two general SFCs interworking with each other, SFC (instead of SSC) is mentioned here.

- To manage the registration of the SFC catalogue in other SFC domains to be interworked with. The registration information includes (but is not limited to) SFC domain identification, SFC identification, SFC name, service functions for this SFC, SFC functionality description, the interworking function to be contacted, etc.
- To construct a combined service chain (or combined SFP) across different SFC domains that may be deployed in the same geographical area but are managed by different service providers. For example, a SFC controller can combine a service chain in a SFC domain with a service security chain in a SSC domain. This can be implemented via a new logic function, i.e., the *interworking function*.
- To update registration information for the SFC/extended SSC controller together with SFC catalogue to be interworked with. There are two possible mechanisms to update registration information, i.e., pull/get updates and push updates. For example, the SFC controller can

get updated registration information of the extended SSC catalogue periodically from the SSC domain. Of course, the extended SSC controller in the SSC domain can report its registration status together with the extended SSC catalogue to the SFC controller in the SFC domain when there is any update.

- To configure the *interworking function* to discover SFC domains to be interworked with. For example, the SFC controller configures *Interworking Function1* to discover the SSC domain.
- To construct both the general SFP in [IETF RFC 7665] and the interworking SFP defined in this Recommendation. For example, the extended SSC controller in an SSC domain can construct an interworking SFP and configures *Interworking Classifier2* to support SSC interworking with SFC.

The new *interworking function* logical function has the following capabilities:

- To support to discovery of SFC domains to be interworked with. For example, *Interworking Funtion1* should have the capability to discover the SSC domain for security services. Related registration and discovery mechanisms are out scope of this Recommendation.
- To support the interworking network service header (NSH) defined in Annex A.1.
- To convert a general NSH defined in [IETF RFC 8300] to an interworking NSH defined in Annex A.1.
- To convert an interworking NSH defined in Annex A.1 to a general NSH defined in [IETF RFC 8300].
- To manage the mapping of interworking SFCs between two SFC domains.
- To have the capabilities of SFF.

The extended classifier i.e., *Interworking classifier* has the following capabilities in addition to the capabilities defined in [IETF RFC 7665]:

- To judge if it is an interworking SFC or a general SFC in [IETF RFC 7665] according to the flag field *"IW"* of NSH.
- To manage both general SFPs and interworking SFPs.
- To respond to the *interworking function* that sends the SFC interworking request. For example, *Interworking Function1* of SFC domain sends a SFC interworking request to the SSC domain for security service. After security service offered in SSC domain, *Interworking Classifier2* in SSC domain should forward the response to *Interworking Function1* in the SFC domain.
- To be deployed with the *interworking function* in the same host to simplify the procedures.

NOTE – When a SSC controller is located in a SFC controller, there are still two separate logical domains, i.e., SFC domain and SSC domain.

8 Procedures for security service chain creation

8.1 Procedures for creating stand-alone SSC

According to Figure 7-1, the procedures for creating a stand-alone SSC are shown in Figure 8-1.

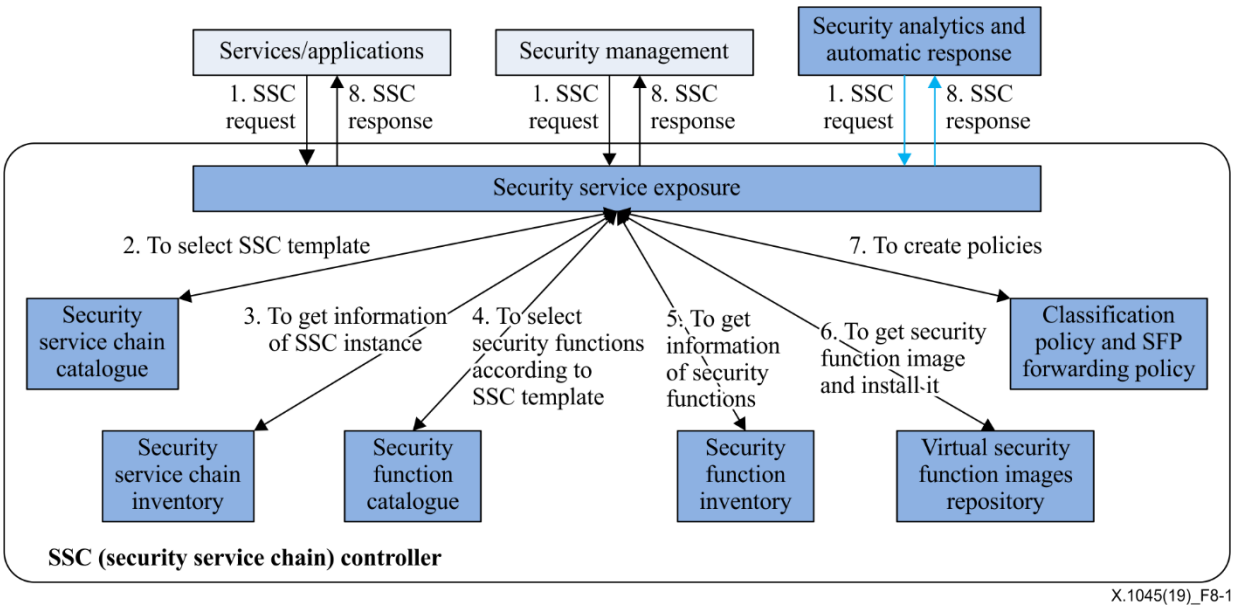


Figure 8-1 – Procedures for creating a stand-alone SSC

The procedures for creating a stand-alone SSC as in Figure 8-1 are described as follows:

1) *Services/applications*, or *security management*, or *security analytics and automatic response (SAAR)* requests customized security protection from *SSC controller* and sends a security service chain (SSC) request to the *SSC controller* according to their security requirements.

2) After receiving a SSC request from *services/applications*, or *security management*, or *security analytics and automatic response (SAAR)*, *security service exposure* looks up the *security service chain catalogue* and selects an appropriate SSC template (i.e., an ordered set of security functions).

3) *Security service exposure* checks if an instance of such a security service chain is registered and active in the *security service chain inventory*. If yes, this active instance of SSC is selected and reused. Then it goes to step 7.

4) If there is no active instance of SSC, *security service exposure* looks up the *security function catalogue* and selects corresponding security functions according to selected SSC template (i.e., an ordered set of security functions in step 2.

5) According to selected security functions, *security service exposure* checks if instances of these security functions are registered and active in the *security function inventory*. If yes, these active instances are selected and reused for this security service chain. An instance of the SSC meeting the received SSC request is created. Then it goes to step 7.

6) If there is no active instance of selected security functions, it looks up a *virtual security function images repository* and selects security functions images which will be instantiated and deployed in the appropriate hosts with the network topology provided by the SDN controller. The instances of selected security functions are created, then an instance of required SSC is created. Corresponding information should be reflected into the *security service chain inventory* (e.g., this newly created SSC) and the *security function inventory* (e.g., newly instantiated security functions).

7) Corresponding SFPs, classification policy and SFP forwarding policy are created and reflected into the classification policy table of the *classifier* and SFP forwarding table of *SFF* respectively.

The implementation of SSC based on SDN can refer to [ITU-T X.1043].

8) The *SSC controller* sends the SSC response message to *services/applications*, or *security management*, or *security analytics and automatic response (SAAR)* to indicate that a customized security service chain is created.

8.2 Procedures for SSC interworking with SFC

It is assumed that the delivery of end-to-end services/applications requires SFC interworking with SSC in order to get security protection. It is also assumed that the components (i.e., *interworking agent*, *interworking function* and *extended classifier*) in Figure 7-2 are ready to provide services of SSC interworking with SFC.

According to Figure 7-2, the procedures for SSC interworking with SFC are shown in Figure 8-2.

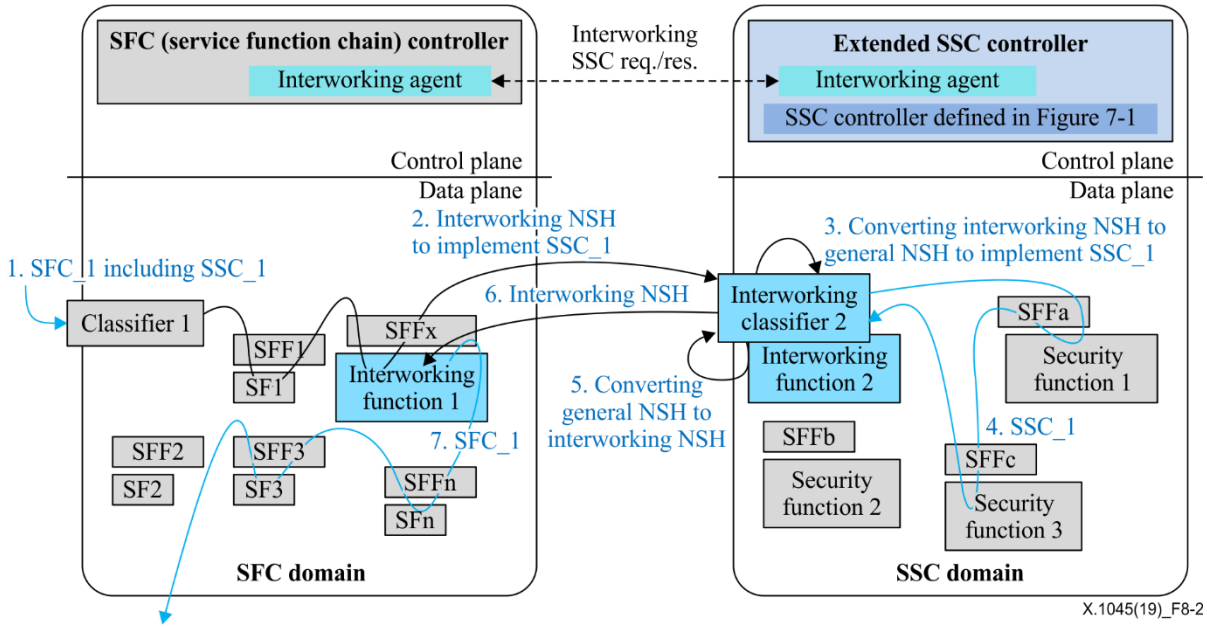


Figure 8-2 – Procedures for SSC interworking with SFC

The procedures for SSC interworking with SFC in Figure 8-2 are described as follows:

Prerequisites: When receiving a SFC request from *services/applications*, the *SFC controller* creates a service chain (e.g., SFC_1) including a SSC request which needs to interwork with SSC domain. Then related classification policy will be reflected into *Classifier1* and related SFP forwarding policy will be reflected into a corresponding *Interworking Function1* and *SFFs* in the SFC domain. How the *SFC controller* creates a service chain SFC_1 is out of the scope of this Recommendation.

1) In the SFC domain, a flow/packet passes through *Classifier1* (e.g., with selecting service chain SFC_1 together with SSC_1) and corresponding *SFFs* (e.g., SFF1) and finally reaches to *Interworking Function1*.

2) After receiving the flow/packet, *Interworking Function1* constructs an interworking NSH as specified in Annex A.1, stores mapping information of the service chain SFC_1 interworking with SSC domain (e.g., interworking flow ID in SFC domain, SPI and SI in SFC domain, SSC domain ID, IP address of *Interworking Classifier2* and *Interworking Function2* in SSC domain, metadata in fixed length context header 1 and 2), then forwards the flow/packet with interworking NSH to *Interworking Classifier2* and *Interworking Function2* in SSC domain. It is assumed that *Interworking Classifier2* and *Interworking Function2* are deployed at the same host.

3) When receiving the flow/packet with interworking NSH from *Interworking Function1* in SFC domain:

a) *Interworking Classifier2* in SSC domain recognizes that this is an interworking SSC request and selects a corresponding interworking security service chain (e.g., SSC_1) according to the metadata in *Fixed Length Context Header2* as defined in Annex A.1. The only one difference between interworking SSC and stand-alone SSC described in clause 8.1 is that the last hop of interworking SSC is *Interworking Classifier2*. If *Interworking Classifier2* finds there is no appropriate or active interworking SSC for this flow/packet, it communicates with the extended *SSC controller* to create an interworking SSC. How the extended *SSC controller* creates such an interworking SSC is described in clause 8.1.

b) *Interworking Function2* in SSC domain stores mapping information of the SSC interworking with SFC (e.g., interworking flow ID in SSC domain, SPI and SI in SSC domain, SFC domain ID, interworking flow ID in SFC domain, SPI and SI in SFC domain, IP address of *Interworking Function1* in SFC domain, metadata in *Fixed Length Context Header 1 and 2*) converts the interworking NSH to a general NSH defined in [IETF RFC 8300] and keeps the metadata in *Fixed Length Context Header 1 and 2*.

c) *Interworking Classifier2* forwards this flow/packet with a general NSH to next hop (e.g., SFFa) in SSC domain according the corresponding SFP.

4) This flow/packet with a general NSH passes through the required security functions and *SFFs* (e.g., SFFa and SFFc) in the SSC domain, then returns back to *Interworking Classifier2* according to the SFP.

5) When receiving this flow/packet with a general NSH from *SFF* in SSC domain:

a) *Interworking Classifier2* recognizes that is an interworking SSC.

b) *Interworking Function2* looks up the mapping information and finds that next hop of this SSC is *Interworking Function1* in SFC domain. *Interworking Function2* converts the general NSH to an interworking NSH.

c) *Interworking Classifier2* forwards this flow/packet with an interworking NSH to *Interworking Function1* in SFC domain.

6) *Interworking Function1* in SFC domain converts interworking NSH to a general NSH and forwards it to next hop (e.g., SFFn) of service chain SFC_1 in SFC domain. At this time, SFC interworking with SSC is completed.

7) The flow/packet goes through the rest path of service chain SFC_1.

9 Customized security services provided based on SSC

9.1 Security service chain for data services based on data labelling

Currently the datacentre industry has many security solutions such as label security [b-LabelSEC], to provide security protection for both data at rest and data in use. It is also very important for data service providers (DSPs) to provide customized security protection (e.g., some data needs only integrity protection during transportation, while other data needs both integrity and confidentiality protection during transportation) for data in transit, such as data transportation from collecting point to datacentre, data transportation within/across datacentres. Although some security mechanisms like virtual private networks (VPNs) have been used to provide security protection for data in transit, all data with different security requirements is protected with the same security level (e.g., same security algorithm, same encryption key, same integrity key, same security filtering policy, etc.), which could not meet the security requirements for highly sensitive data.

With the data labelling scheme described in Annex B, customized security services could be offered for the labelled data when it is imported from data collecting points to datacentres, transported between datacentre's network perimeter and data storage devices, transported across geo-distributed

datacentres or various sub-systems, as well as stored in the storage. Moreover, labelled data can be quickly routed to appropriate cloud-based databases (e.g., rational database, MongoDB, HBase, etc.) so that system performance in datacentres will be improved.

9.1.1 Collecting data and storing data in datacentres securely

Figure 9-1 shows how to create stand-alone security service chains for the data which is collected and imported into the datacentre, then securely stored in a cloud storage device.

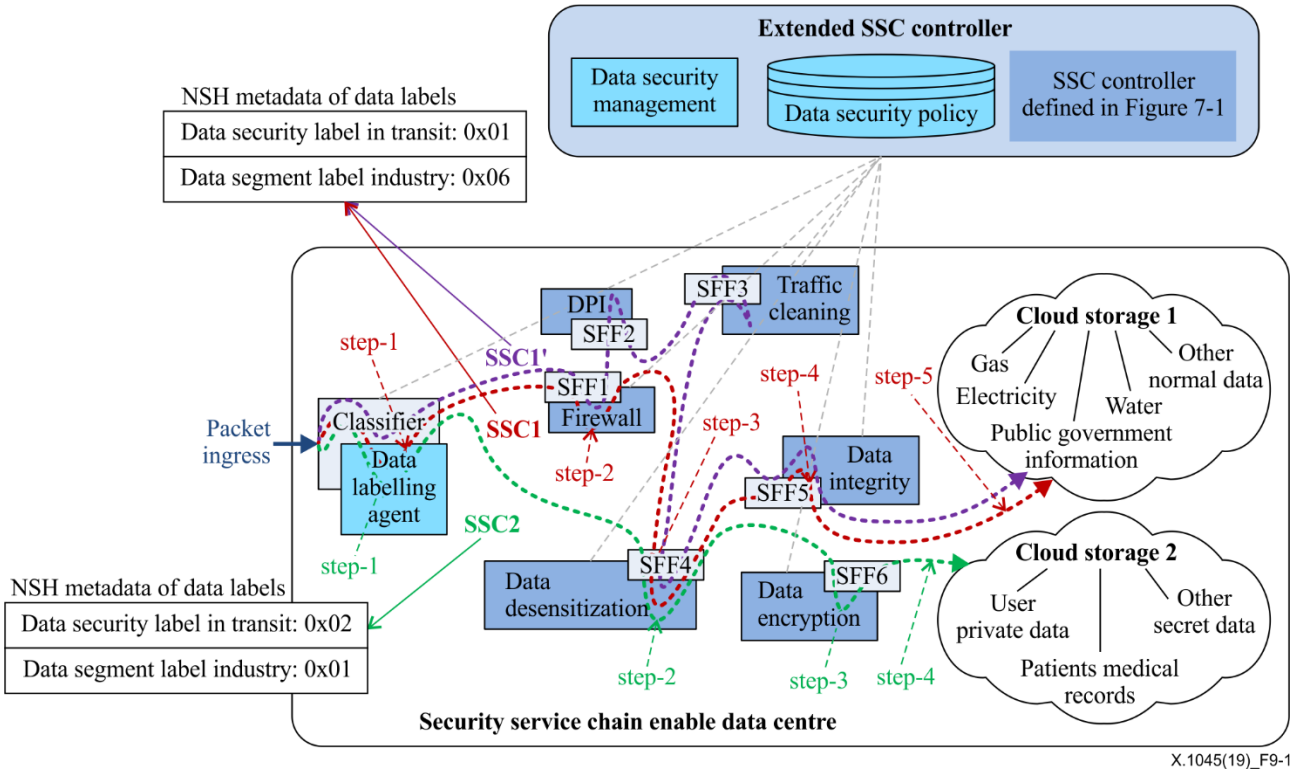


Figure 9-1 – Data labelling and secure storage in datacentre

In order to support the data labelling scheme in Annex B, the SSC controller in Figure 7-1 should have the capabilities to define fine-grained data security policy according to a data labelling scheme, to create classification policies and SFP forwarding policies based on data labelling. So, the SSC controller in Figure 7-1 will be extended with the two new logical functions in light blue blocks i.e., *data security management* (i.e., the activities associated with controlling and protecting access to data, event monitoring, reporting, policy and auditing) and *data security policy* (i.e., a type of security policy that aims to design, implement, monitor and manage security over the data, an example is defined in Annex B.1). The SFFs in Figure 7-1 should be extended and have the capabilities to support extended NSH based on data labelling in Annex A.2. The classifier in Figure 7-1 should be extended and have the capabilities to add data labels to the incoming data with the logic function *data labelling agent*, to generate an extended NSH based on data labelling in Annex A.2.

When the logical function *data labelling agent* receives data packets, it should do one of three operations as follows:

- If the received data packets have no data labels, data labels are generated and added into NSH metadata, database metadata and payload metadata;
- If the received data packets have data labels defined by the same data labelling scheme as defined in this Recommendation, all data labels are correspondingly kept as they are in NSH metadata, database metadata and payload metadata;

- If the received data has data labels defined by other data labelling systems such as label security [b-LabelSEC], the existing data labels will be converted into data security labels for data in transit then added into NSH metadata since existing data label systems do not define data labels for data in transit; the existing data labels will be mapped into data security level labels defined in this Recommendation.

In Figure 9-1, the security service chain SSC1 (*classifier with data labelling agent ->SFF1 -> Firewall ->SFF1 ->SFF4 -> data desensitization ->SFF4 ->SFF5 -> data integrity ->SFF5 -> cloud storage1)* shows one possible method to protect energy/utilities data when it is collected and imported into the datacentre. The procedures for SSC1 (i.e., step-1 to step-5 in red text) are described below:

step-1) When energy/utilities data is being imported into the datacentre, the logical function *data labelling agent* at the datacentre's network perimeter adds appropriate data labels to the energy/utilities data. According to data labels and classifier policies, the logical function *Classifier* inserts extended NSH (with metadata of data labels in Figure 9-1) to the packet/flow and chooses SFP for data.

step-2) Energy/utilities data is scanned by security function *Firewall* (e.g., to guarantee that the energy/utilities data source is allowed to upload the data).

step-3) User's sensitive information (e.g., user name, home address) included in energy/utilities data is removed by security function *data desensitization* (e.g., to guarantee that user privacy is protected).

step-4) An integrity protection code is generated and added by security function *data integrity* (e.g., to guarantee that data at rest is not modified by illegal user or application, in this way the bill of utilities will be generated correctly).

step-5) SFF5 removes the extended NSH. Desensitized energy/utilities data with integrity protection code is stored in *cloud storage1*.

An alternative security service chain of SSC1 is SSC1' *(Classifier with data labelling agent ->SFF1 -> Firewall ->SFF1 -> SFF2 -> DPI -> SFF2 -> SFF3 -> traffic cleaning -> SFF3 ->SFF4 -> data desensitization ->SFF4 ->SFF5 -> data integrity ->SFF5 -> cloud storage1)*. The difference between SSC1 and SSC1' is that unwanted data including virus or malware should be removed before storing energy/utilities data in *"cloud storage1"*. Security service chain SSC1' is mentioned in this Recommendation in order to explain that classification policy based on data labels together with extended NSH can work well with other classification policies without any impact.

In Figure 9-1, security service chain SSC2 (*Classifier with data labelling agent -> SFF4 -> data desensitization -> SFF4-> SFF6 -> data encryption -> SFF6 -> cloud storage2)* shows one possible method to protect healthcare/pharma data when it is collected and imported into the datacentre and stored in the datacentre. End users' medical records have to be desensitized and stored as encrypted data in order to improve users' privacy protection. The procedures for SSC2 (i.e., step-1) to step-4) in green text) are outlined as below:

step-1) When healthcare/pharma data is collected and imported into the datacentre, logical function *data labelling agent* at the datacentre's network perimeter adds appropriate data labels to healthcare/pharma data. According to data labels and other classifier policies, logical function *Classifier* inserts extended NSH (with metadata of data labels in Figure 9-1) to the packet/flow and chooses SFP for data.

step-2) End user's medical records and sensitive information (e.g., name, home address) are split into different parts by security function *data desensitization* to support isolated storage so that end users' privacy protection can be improved.

step-3) Different parts of healthcare/pharma data are encrypted by security function *data encryption* separately to guarantee that the data at rest is accessed only by authorized user or application.

step-4) SFF6 removes the extended NSH. Different encrypted parts of healthcare/pharma data are stored in *cloud storage2* separately in logical isolation or physical isolation. In this way, the desensitized data related to medicines and/or medical instruments can be shared with the manufacturers to make appropriate product plans.

9.1.2 Data sharing with third parties securely

Figure 9-2 shows how to securely share healthcare/pharma data with third parties with user's consensus, in order to enable making product plans, analysing market trends, creating value-added services.

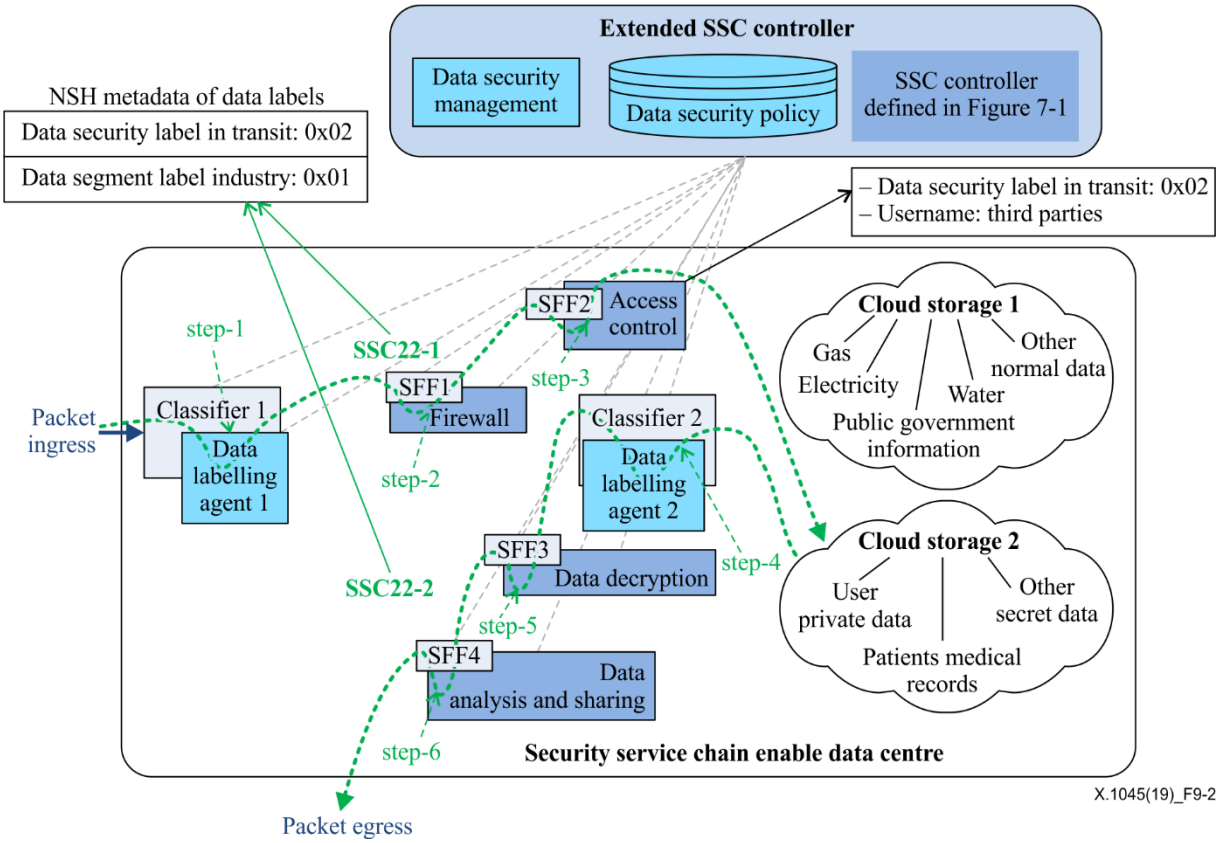


Figure 9-2 – Healthcare/pharma data shared with third parties

In Figure 9-2, security service chains SSC22-1 (*Classifier1 with data labelling agent1 -> SFF1 > Firewall-> SFF1-> SFF2 -> access control -> SFF2 -> Cloud Storage2*) and SSC22-2 (*Classifier2 with data labelling agent2-> SFF3 ->data decryption -> SFF3 -> SFF4 -> data analysis and sharing-> SFF4*) shows one possible method to securely share healthcare/pharma data with third parties. The procedures for SSC22-1 (i.e., *step-1 to step-3 in green text*) and SSC22-2 (i.e., *step-4 to step-6 in green text*) are described as below:

step-1) After receiving data access request, logical function *data labelling agent1* at datacentre's network perimeter analyses and decides what data labels are applicable to this request, and then logical function *Classifier1* inserts extended NSH (with metadata of data labels in Figure 9-2) to the packet/flow and chooses SFP for the request.

step-2) The data access request is scanned by security function *Firewall* (e.g., to guarantee that the request is valid and does not including virus or malware).

step-3) The data access request is checked by security function *access control* (e.g., to guarantee that the third party has the right to access the requested healthcare/pharma data). Then SFF2 removes the extended NSH and forwards the data access request to cloud storage2.

step-4) After getting data from cloud storage, logical function *data labelling agent2* near to cloud storage analyses and decides what data labels are applicable to this response, and then logical function *Classifiers* inserts extended NSH (with metadata of data labels in Figure 9-2) to the packet/flow and chooses SFP for the response.

step-5) Data decryption is done by the security function *data decryption*.

step-6) Healthcare/pharma data is analysed by the service function *data analysis and sharing*. SFF4 removes the extended NSH, then responds to the data access request.

9.2 Security service chain for ITS services

Special vehicle (SV) speedup service described in Annex C.1 is one of the ITS services. There may be an attacker who impersonates a legal user to access an SV speedup service. A virus or malware is also possible to be injected into the request which is sent to the *urban traffic management service* to control the traffic lights. So, it is very important to enforce security policies for ITS services, such as the authentication of the user with the device (i.e., ITS station), attacks detection and traffic cleaning for the request to the *urban traffic management service*, etc. One possible deployment of the security service chain for ITS service is described in [b-IWCMC].

In order to elaborate SSC interworking with SFC, it is assumed that the ITS SFC domain and the SSC domain are managed by different service providers. On the other hand, the SSC domain can be interworked with other SFC domains except ITS SFC domain. It is also assumed that the underlying network of these two SFC domains can support SFC.

9.2.1 Authentication

In order to mitigate/prevent impersonation attacks, it is very important to authenticate the user before he/she accesses the SV speedup service. In order to describe how a SSC domain interworks with a SFC domain, it is assumed that the authentication service is provided by a third trusted party (e.g., government, public safety department, etc.). Of course, an authentication service can also be deployed in the ITS SFC domain.

The service chain ITS SFC1 in SFC domain described in Annex C.1 interworks with the service chain SSC1 in the SSC domain to provide authentication of the user for a SV speedup service as shown in Figure 9-3.

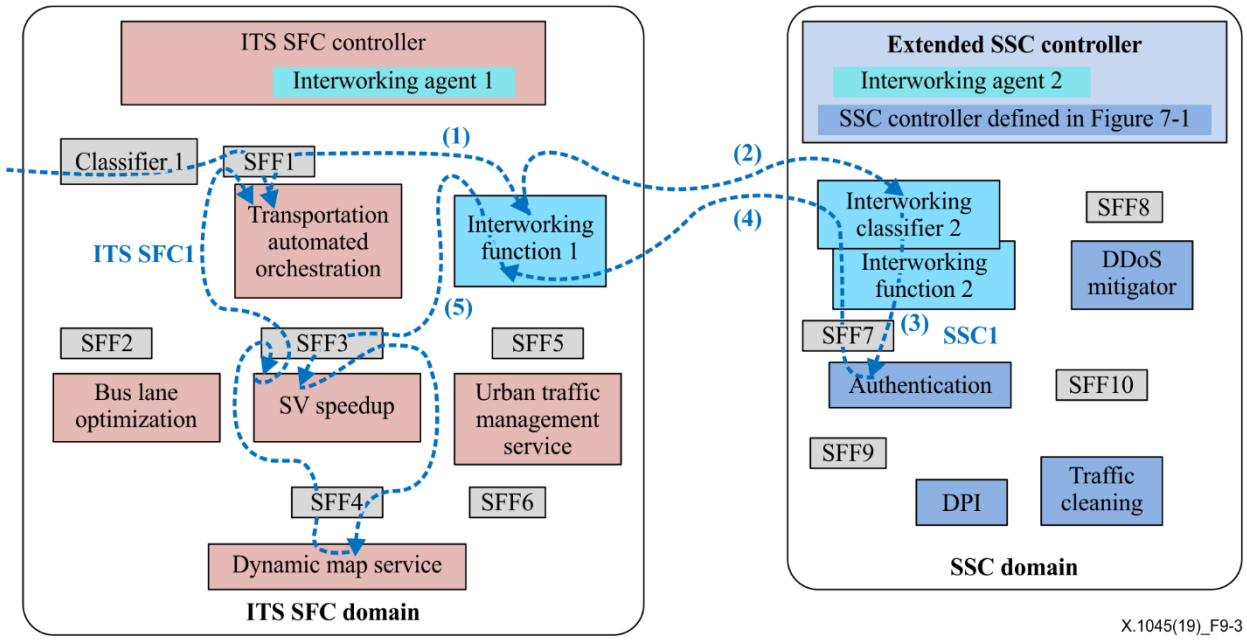


Figure 9-3 – ITS SFC1 in SFC domain interworking with SSC1 in SSC domain for authentication

The combined service chain ITS_SFC1-SSC1, which shows the service chain ITS SFC1 in SFC domain interworking with the service chain SSC1 in SSC domain, is described as follows: *Classifier1 -> SFF1 -> Transportation Automated Orchestration -> SFF1 -> Interworking Function1 -> Interworking Classifier2 and Interworking Function2 -> SFF7 -> Authentication -> SFF7 -> Interworking Classifier2 and Interworking Function2 -> Interworking Function1 -> SFF3 -> SV Speedup -> SFF3 -> SFF4 -> Dynamic Map Service -> SFF4 -> SFF3 -> SV Speedup -> SFF3 -> SFF1 -> Transport Automated Orchestration*.

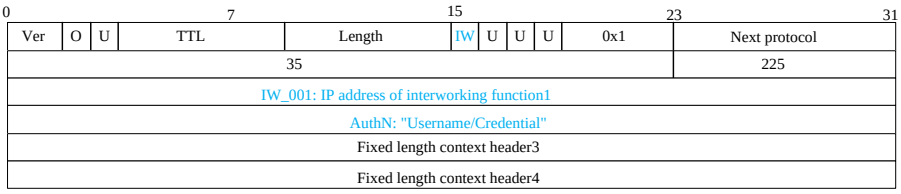
Before describing key procedures for the combined service chain ITS_SFC1-SSC1 in Figure 9-3, some prerequisites are listed as below:

- It is assumed that *Interworking Classifier2* and *Interworking Function2* are deployed on the same host.
- *Interworking Agent1* in ITS SFC controller manages the registration of security functions in SSC domain.
- According to ITS security requirements, *Interworking Agent1* in ITS SFC controller manages logical security service chains, such as SSC1 (*Authentication*) and SSC2 (*DPI -> Traffic cleaning*).
- *Interworking Agent1* in ITS SFC controller configures *Interworking Function1* to support a SFC domain interworking with SSC domain. Interworking functions manages the mapping of the service chain in the SFC domain interworking with the service chain in SSC domain.
- As for the updates of registration information for SFC domains together with SSCs to be interworked with, *Interworking Agent1* in ITS SFC controller can get updated registration information of security functions periodically from the SSC domain, or *Interworking Agent2* in extended SSC controller reports its registration status when there is any update.

Some key procedures for the combined service chain ITS_SFC1-SSC1 in Figure 9-3 are described below:

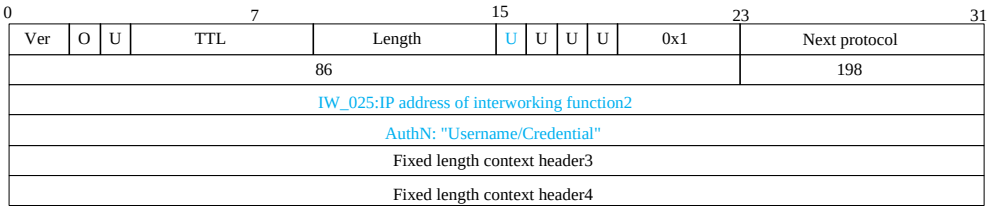
1) When receiving incoming flow/packet from *transport automated orchestration*, *Interworking Function1* constructs an interworking NSH as below:

NOTE – The value of other fields not related to SFC interworking is not shown.



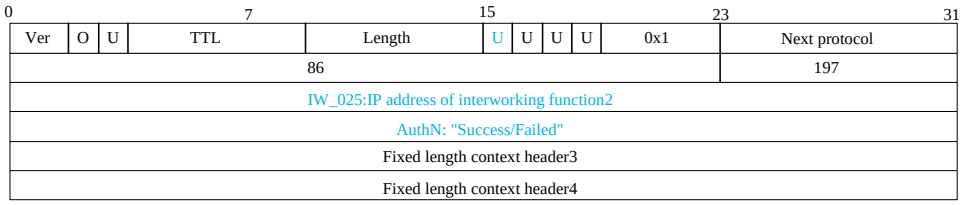
Interworking Function1 stores the mapping information and forwards the flow/packet with interworking NSH to *Interworking Classifier2* and *Interworking Function2*.

2) *Interworking Classifier2* receives the message from *Interworking Function1*. According to interworking flow ID, *Interworking Classifier2* chooses the service chain SSC1 as well as indicating that the SF (i.e., Authentication) should return the flow/packet to *Interworking Classifier2* and *Interworking Function2*. *Interworking Function2* converts the interworking NSH to general NSH defined in [IETF RFC 8300] as below:



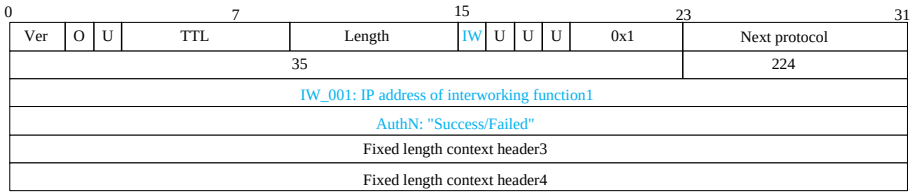
Interworking Classifier2 and *Interworking Function2* store the mapping information and forward the flow/packet with general NSH defined in [IETF RFC 8300] to the SF i.e., *Authentication*.

3) *Authentication* authenticates the user with "AuthN: Username/Credential" then updates the field "Fixed Length Context Header2" with "AuthN: Success/Failed". The NSH was updated as below:



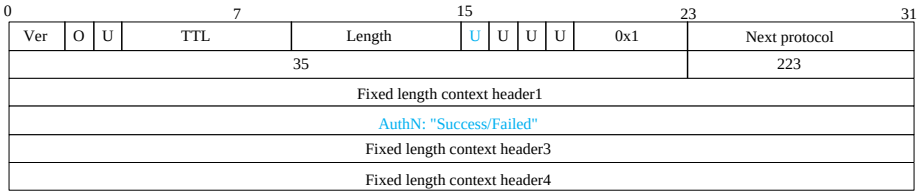
Authentication checks the field Fixed Length Context Header1 and finds the address of *Interworking Classifier2* and *Interworking Function2*, then forwards the flow/packet to *Interworking Classifier2* and *Interworking Function2*.

4) *Interworking Classifier2* and *Interworking Function2* convert general NSH to interworking NSH as below:



Interworking Classifier2 and *Interworking Function2* check the mapping information and find that next hop is *Interworking Function1* in the SFC domain, then forward the flow/packet with interworking NSH to *Interworking Function1*.

5) *Interworking Function1* receives the message from *Interworking Classifier2* and *Interworking Function2* in SSC domain, and converts interworking NSH to general NSH defined in [IETF RFC 8300] as below:



Interworking Function1 checks the mapping information and finds that next hop is *SV Speedup*, then forwards the flow/packet with general NSH to *SV Speedup* via *SFF3*.

9.2.2 Traffic cleaning

According to the ITS SFC2 described in Annex C.1, *transport automated orchestration* checks congestion status in this route continually. If the congestion is high, it sends a speedup road side unit (RSU) such as traffic lights and velometer command to the *urban traffic management service* to control the traffic lights to be on green or reduce the red time to reduce the congestion in the route.

However, there may be a virus or malware injected by the attackers into the speedup RSU command to the *urban traffic management service*, which is very important for the urban traffic. So, it is necessary to do virus or malware scanning, detecting and mitigating for the flow/packet which carries the speedup RSU command before this command is forwarded to *urban traffic management service*. This can be achieved by combining the service chain ITS SFC2 with the service chain SSC2 in Figure 9-4.

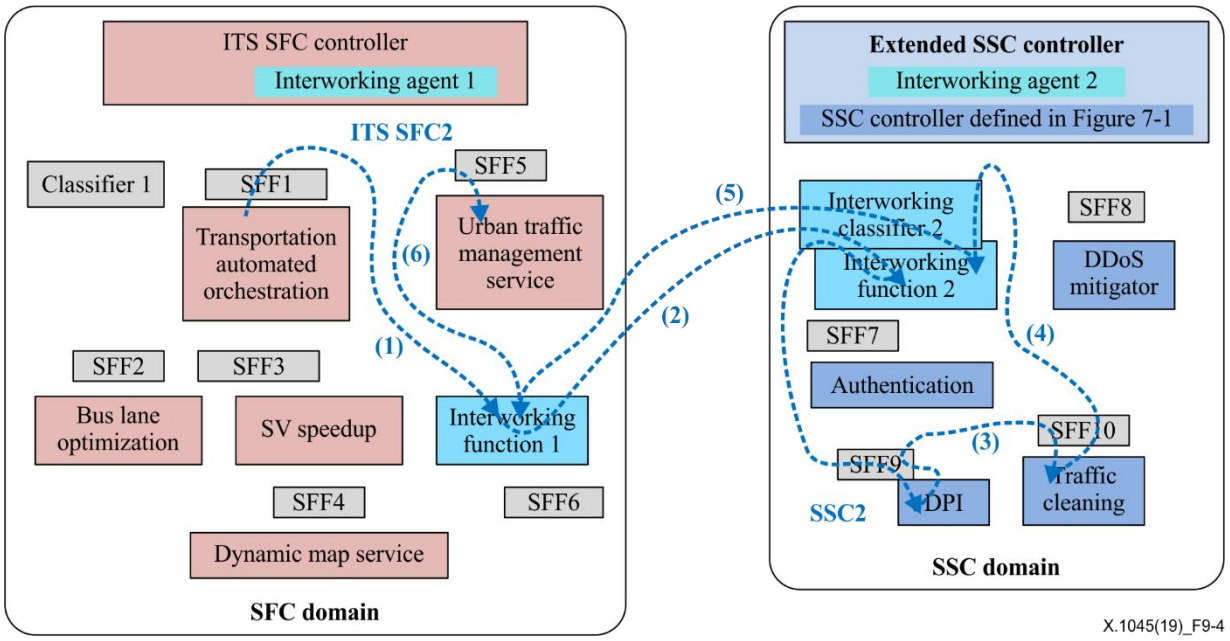


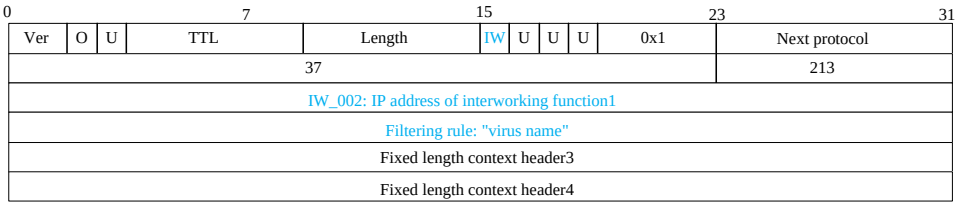
Figure 9-4 – ITS SFC1 in SFC domain interworking with SSC2 in SSC domain for traffic cleaning

The combined service chain ITS_SFC2-SSC2, which shows the service chain ITS SFC2 interworking with the service chain SSC2, is described as following: *Transport Automated Orchestration -> SFF1 -> Interworking Function1 -> Interworking Classifier2 and Interworking Function2 -> SFF8 -> DPI -> SFF8 -> SFF9 -> Traffic Cleaning -> SFF9 -> Interworking Classifier2 and Interworking Function2 -> SFC Interworking Function1 -> SFF5 -> Urban Traffic Management Service.*

Some key procedures for the combined service chain ITS_SFC2-SSC2 in Figure 9-4 are described below:

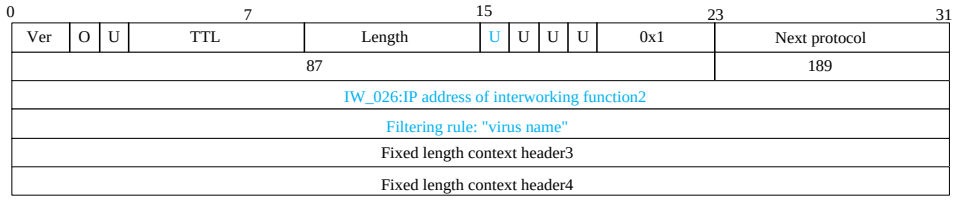
1) When receiving incoming flow/packet from *transport automated orchestration*, *Interworking Function1* constructs an interworking NSH as below:

NOTE – The value of other fields not related to SFC interworking is not shown.



Interworking Function1 stores the mapping information and forwards the flow/packet with interworking NSH to *Interworking Classifier2* and *Interworking Function2*.

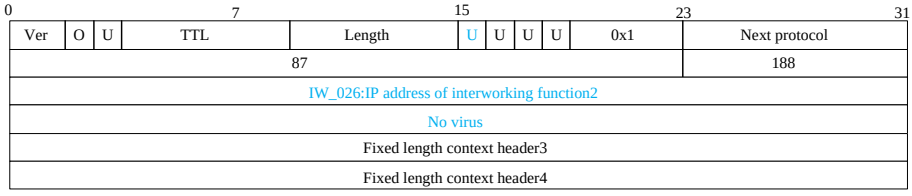
2) *Interworking Classifier2* in SSC domain receives the message from *Interworking Function1* in SFC domain. According to the interworking flow ID, *Interworking Classifier2* chooses SSC2 as well as indicating that the SF (i.e., DPI or traffic cleaning) should return the flow/packet to *Interworking Classifier2* and *Interworking Function2*. *Interworking Function2* converts the interworking NSH to general NSH defined in [IETF RFC 8300] as shown below:



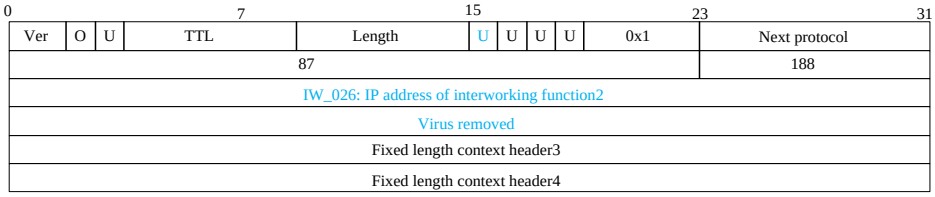
Interworking Classifier2 and *Interworking Function2* store the mapping information and forward the flow/packet with general NSH defined in [IETF RFC 8300] to the DPI.

3) According the *Filtering Rule: "Virus name"*, the *DPI* scans the flow/packet and detects if there is any virus in the flow/packet. If there is no virus, *DPI* forwards the flow/packet to *Interworking Classifier2* and *Interworking Function2*. If there is a virus, *DPI* forwards the flow/packet to *traffic cleaning*. Then the virus is removed from the packet by *traffic cleaning*. After that, *traffic cleaning* forwards the cleaned flow/packet to *Interworking Classifier2* and *Interworking Function2*. Before the *DPI* or *traffic cleaning* forwarding of the flow/packet, the NSH was updated as below:

NSH updated by DPI

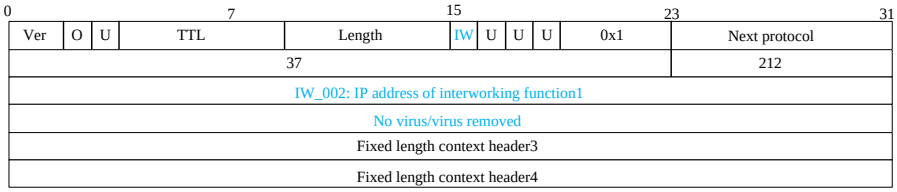


NSH updated by traffic cleaning



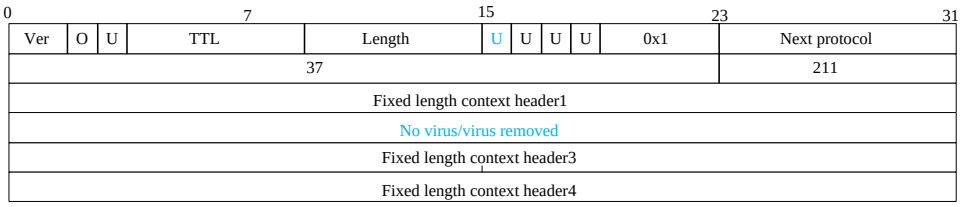
DPI or *traffic cleaning* checks the field Fixed Length Context Header1 and finds the address of *Interworking Classifier2* and *Interworking Function2*, then forwards the flow/packet to *Interworking Classifier2* and *Interworking Function2*.

(4) *Interworking Classifier2* and *Interworking Function2* convert general NSH to interworking NSH as below:



Interworking Classifier2 and *Interworking Function2* check the mapping information and find that next hop is *Interworking Function1* in SFC domain, then forward the flow/packet with interworking NSH to *Interworking Function1* in the ITS SFC domain.

(5) *Interworking Function1* receives the message from *Interworking Classifier2* and *Interworking Function2* in SSC domain, and convert interworking NSH to general NSH defined in [IETF RFC 8300] as below:



(6) *Interworking Function1* forwards the flow/packet with general NSH to the urban traffic management service.

9.3 Security service chain to enable mitigating/preventing of network attacks automatically

Generally, there are three steps to prevent or mitigate network attacks: 1) detecting network attacks; 2) tracing network attacks to their sources; and 3) blocking network attacks.

How to trace network attacks to their sources in SFC overlay network with higher performance and the how to block those attacks based on security service chain will be described in this Recommendation. However, how to detect network attacks (e.g., with IDS) is out of scope of this Recommendation.

Denial of service (DoS) attack is very common and is one of the top 7 network attacks reviewed in [b-Threat Report]. So, how to mitigate/prevent DoS attacks automatically is taken as an example and elaborated in this Recommendation.

9.3.1 Tracing network attacks to their sources in a single SFC domain and blocking them automatically

Figure 9-5 shows a service chain SFC_t1 as follows: Classifier ->SFF1 ->SF2 -> SFF1 -> SFF2 -> SF3 ->SFF2 -> IDS ->Server.

It is assumed that the attacker hijacked the service function (i.e., SF2 in Figure 9-5(a) and Figure 9-5(b)) to create intentional DoS attacks to prevent legitimate users from accessing targeted services (i.e., server (victim) in Figure 9-5).

It is also assumed that both the SFC controller and SSC controller defined in Figure 7-1 are deployed in the same host in order to simplify the description.

In order to block/prevent the DoS attacks, firstly it is necessary to identify the source of DoS attacks, i.e., SF2 in Figure 9-5(a) and Figure 9-5(b). Then, the organization or the host or the person who originated the attack can be identified. Finally, the source of DoS attacks (i.e., SF2 in

Figure 9-5) will be isolated from the network and also be removed from the SFP. That is to say, a new SFP will be created and the attacked service function (i.e., SF2 in Figure 9-5) will be replaced by another normal and secure one (i.e., SF20 in Figure 9-5). In this way, the attacks from SF2 are blocked. All of this is shown in Figure 9-5. The only difference between Figure 9-5(a) and Figure 9-5(b) is the deployment of IDS.

- Figure 9-5(a): in-path deployment of IDS;
- Figure 9-5(b): traffic redirection deployment of IDS.

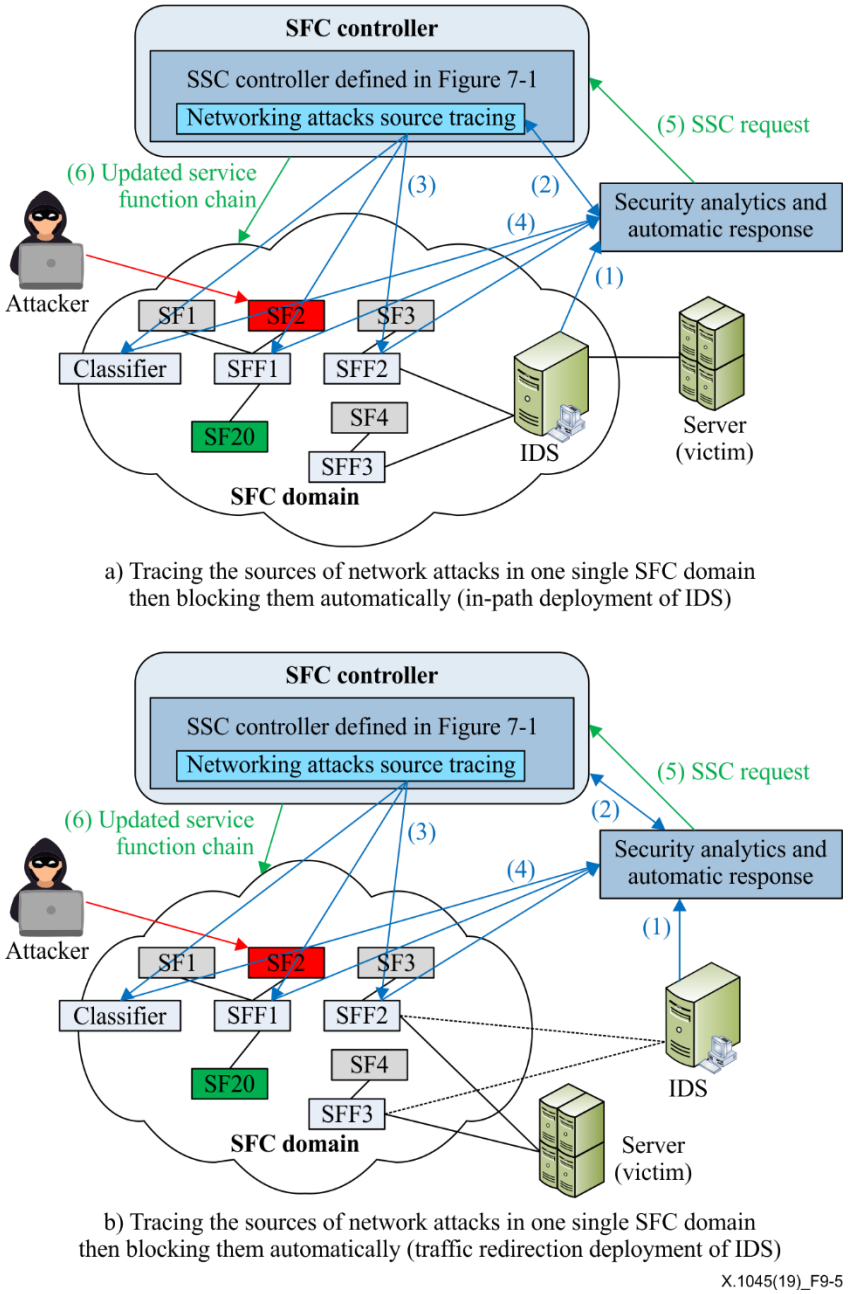


Figure 9-5 – Tracing network attacks to their sources in a single SFC domain and blocking them automatically

Before starting to describe the procedures in Figure 9-5, it is assumed that the classifier, SFFs and IDS have the capabilities to recognize the extended NSH defined in Annex A.3 in order to support tracing network attacks to their sources; to record/log the transmitted flow/packet features (e.g., SPI, classifier ID, SFF ID, SFC ID, the source IP address, the destination IP address, corresponding hash values of the above information together with part of payload, etc.); to insert the flow/packet

features into the extended NSH as defined in Annex A.3. It is assumed that IDS in Figure 9-5(a) and Figure 9-5(b) have the capability of SFF.

The procedures for tracing network attacks to their sources in a single SFC domain and then blocking them automatically are described as follows:

(1) When IDS detects DoS attacks, it sends alerts to the logic function *Security analytics and automatic response (SAAR)*. The function *SAAR* logs this event and collects suspicious packet information (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing) from IDS if the packet information is not included in the alert.

NOTE – If IDS is not SFF functioned, the *SAAR* gets SPI from IDS attached SFF.

(2) After receiving the malicious packet information, the function *Security analytics and automatic response* generates tracing-attack-source request with the flow/packet features and sends the request to the logic function "Network attacks source tracing" within SSC Controller. After receiving the tracing-attack-source request with the flow/packet features, the function *"Network attacks source tracing"* within SSC controller responds to *Security analytics and automatic response* with SFP of the service chain SFC_t1 (i.e., Classifier ->SFF1 ->SF2 -> SFF1 -> SFF2 -> SF3 ->SFF2 -> IDS ->Server) according to the SPI identifier.

(3) The function *Network attacks source tracing* within SSC controller notifies the classifier, SFF1, SFF2 and IDS (belonging to the SFP of service chain SFC_t1) to send their packet matching result to the *Security analytics and automatic response* function.

(4) After receiving the tracing-attack-source notification message with the flow/packet features (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing), Classifier and SFF1 and SFF2 and IDS calculate the hash values of the information of the tracing request; then match them with the ones recorded previously in Classifier and SFF1 and SFF2 and IDS respectively, then concurrently send the packet matching results to the *Security analytics and automatic response* function.

(5) Based on the SPI with corresponding SFP from the function *Network attacks source tracing* within SSC controller and based on the flow/packet matching results collected from Classifier and SFF1 and SFF2 and IDS, the function *Security analytics and automatic response* can judge that the attacks are from SFF1 and finally identify that the attacks source is SF2. The function *Security analytics and automatic response* sends an SSC request to the SSC controller and indicates that the service function SF2 is the source of network attacks.

(6) After receiving SSC request from the function *Security analytics and automatic response*, SSC controller isolates/suspends the service function SF2. Then the SSC controller together with the SFC controller to create a new SFP, i.e., Classifier ->SFF1 ->SF20 -> SFF1 -> SFF2 -> SF3 ->SFF2 -> IDS ->Server. In this way, the source of network attacks SF2 is blocked.

9.3.2 Tracing network attacks to their sources across SFC domains and blocking them automatically

Figure 9-6 shows a service chain SFC_t2 as follows: Classifier1->SFF11 ->SF12 -> SFF11 -> SFF12 -> SF13 ->SFF12 -> Classifier2 -> SFF21 -> SF21 ->SFF21 ->SFF22 -> SF22 ->SFF22 -> Classifier4 -> SFF41 ->SF41 ->SFF41 -> IDS ->Server.

It is assumed that the attacker hijacked the service function (i.e., SF12 in SFC domain-1 of Figure 9-6) to create intentional DoS attacks to prevent legitimate users from accessing targeted services (i.e., server (victim) in SFC domain-4 of Figure 9-6). Before blocking/preventing the DoS attacks, it is first necessary to identify the DoS attacks source, i.e., SF12 in Figure 9-6.

It is also assumed that both the SFC controller and the SSC controller defined in Figure 7-1 are deployed in the same host for each SFC domain in Figure 9-6 in order to simplify the description. Moreover, all assumptions for classifier and SFFs and IDS described in clause 9.2.1 are also applied for classifiers and SFFs and IDS described in clause 9.2.2.

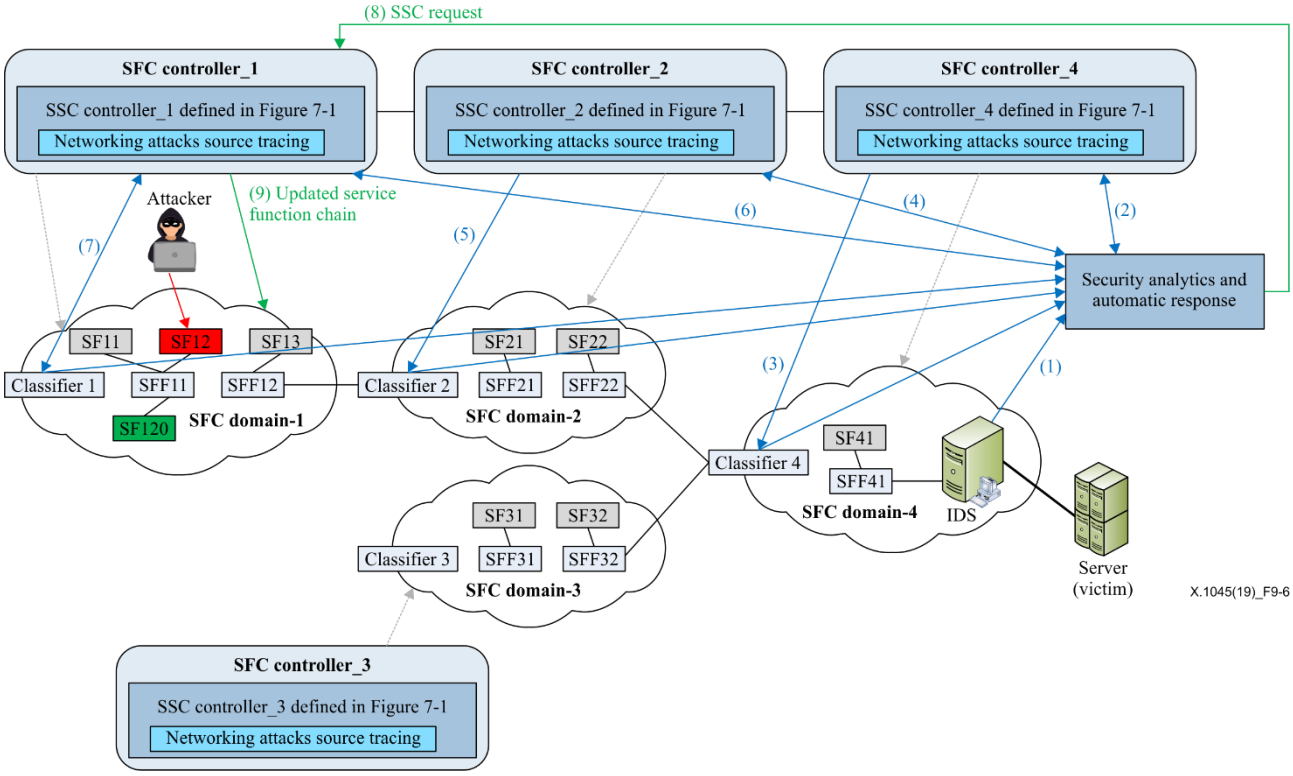


Figure 9-6 – Tracing network attacks to their sources across SFC domains

The procedures for tracing network attacks to their sources across SFC domains and blocking them automatically are described as follows:

(1) When an IDS in the SFC domain-4 detects DoS attacks, it sends alerts to the logic function *Security analytics and automatic response*. The function *SAAR* logs this event and collects suspicious packet information (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing) from IDS if the packet information is not included in the alert.

NOTE – If IDS is not SFF functioned, the SAAR gets SPI from IDS attached.

(2) After receiving the malicious packet information, the function *Security analytics and automatic response* generates the tracing-attack-source request and sends the request to the logic function *network attacks source tracing* within SSC Controller_4. After receiving the tracing-attack-source request with the flow/packet features, the function *network attacks source tracing* within SSC Controller_4 responds to *Security analytics and automatic response* with SFP of service chain in SFC domain-4 (i.e., Classifier4 -> SFF41 ->SF41 ->SFF41 -> IDS ->Server) according to SPI identifier.

(3) The function *Network attacks source tracing* within SSC Controller_4 notifies Classifier4, SFF41 and IDS to send their packet matching result to the function *Security analytics and automatic response*. After receiving the tracing-attack-source notification message with the flow/packet features (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing), Classifier4 and SFF41 and IDS calculate the hash values of the information of the tracing request; then match them with the ones recorded previously in Classifier4 and SFF41 and IDS respectively, then concurrently send the packet matching results to the function *Security analytics and automatic response*.

(4) Based on the SPI with corresponding SFP from the function *Network attacks source tracing* within SSC Controller_4 and based on the flow/packet matching results collected from Classifier4 and SFF41 and IDS, the function *Security analytics and automatic response* finds that this attack already happened before the flow/packet entering into SFC domain-4 and finds that the flow/packet is coming from SFC domain-2. The function *Security analytics and automatic response* sends the tracing-attack-source request to the logic function *Network attacks source tracing* within SSC Controller_2. After receiving the tracing request with the flow/packet features, the function *Network attacks source tracing* within SSC Controller_2 responds to *Security analytics and automatic response* with SFP of service chain in SFC domain-4 (i.e., Classifier2 -> SFF21 -> SF21 ->SFF21 ->SFF22 -> SF22 ->SFF22) according to SPI identifier.

(5) The function *Network attacks source tracing* within SSC Controller_2 notifies Classifier2, SFF21 and SFF22 to send their packet matching result to the function *Security analytics and automatic response*. After receiving the tracing-attack-source notification message with the flow/packet features (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing), Classifier2 and SFF21 and SFF22 calculate the hash values of the information of the tracing request; then match them with the ones recorded previously in Classifier2 and SFF21 and SFF22 respectively, then concurrently send the packet matching results to the function *Security analytics and automatic response*.

(6) Based on the SPI with corresponding SFP from the function *Network attacks source tracing* within SSC Controller_2 and based on the flow/packet matching results collected from Classifier2 and SFF21 and SFF22, the function *Security analytics and automatic response* finds that this attack already happened before the flow/packet entering into SFC domain-2 and finds that the flow/packet is coming from SFC domain-1. The function *Security analytics and automatic response* sends the tracing-attack-source request to the logic function *Network attacks source tracing* within SSC Controller_1. After receiving the tracing request with the flow/packet features, the function *Network attacks source tracing* within SSC Controller_1 responds to *Security analytics and automatic response* with SFP of service chain in SFC domain-4 (i.e., Classifier1->SFF11 ->SF12 -> SFF11 -> SFF12 -> SF13 ->SFF12) according to SPI identifier.

(7) The function *Network attacks source tracing* within SSC Controller_1 notifies Classifier1, SFF11 and SFF12 to send their packet matching result to the function *Security analytics and automatic response*. After receiving the tracing-attack-source notification message with the flow/packet features (e.g., SPI, the source IP address, the destination IP address, part of payload for tracing), Classifier1 and SFF11 and SFF12 calculate the hash values of the information of the tracing request; then match them with the ones recorded previously in Classifier1 and SFF11 and SFF12 respectively, then concurrently send the packet matching results at the same time to the function *Security analytics and automatic response*.

(8) Based on the SPI with corresponding SFP from the function *Network attacks source tracing* within SSC Controller_1 and based on the flow/packet matching results collected from Classifier1 and SFF11 and SFF12, the function *Security analytics and automatic response* can judge that the attacks are from SFF11 in SFC domain-1 and finally identify that the attacks source is SF12. The function *Security analytics and automatic response* sends SSC request to SSC Controller_1 and indicates that service function SF12 is the source of network attacks.

(9) After receiving SSC request from the function *Security analytics and automatic response*, SSC controller_1 isolates/suspends the service function SF12. Then SSC Controller_1 together with SFC controller_1 to create a new SFP within SFC domain-1(i.e., i.e., Classifier1->SFF11 ->SF120 -> SFF11 -> SFF12 -> SF13 ->SFF12). Then the new SFC across SFC domains is: Classifier1->SFF11 ->SF120 -> SFF11 -> SFF12 -> SF13 ->SFF12 -> Classifier2 -> SFF21 -> SF21 ->SFF21 ->SFF22 -> SF22 ->SFF22 -> Classifier4 -> SFF41 ->SF41 ->SFF41 -> IDS ->Server. In this way, the source of network attacks SF12 is blocked.

Annex A

IETF SFC NSH extensions

(This annex forms an integral part of this Recommendation.)

The aim of this annex is to extend IETF SFC NSH [IETF RFC 8300] in order to support SFC features defined in this Recommendation.

In order to simplify the description in this annex, the extension on NSH with MD Type=0x1 (i.e., fixed length context headers) is taken as an example for this document. Of course, the extension is also applied to NSH with MD Type=0x2 (i.e., variable length context headers). Moreover, some security aspects not related to SFC features defined in this Recommendation are omitted in this annex, such as how to provide confidentiality and integrity for NSH during its transportation between SFC domains or even within one SFC domain; how to encapsulate extended NSH with an outer transport encapsulation, etc.

IETF SFC NSH with MD Type=0x1 (i.e., fixed length context headers) defined in [IETF RFC 8300] can be described as follows:

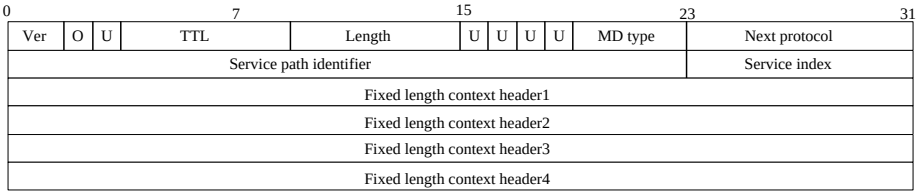


Figure A.1 – IETF SFC NSH with MD Type=0x1

A.1 NSH extensions to support the service chain in one SFC domain interworking with another service chain in another SFC domain

In Figure A.1, there are five unassigned bits marked as "U" in the NSH base header. One of them can be used to indicate whether this NSH is a general NSH as defined in [IETF RFC 8300] or an interworking NSH as defined in this Recommendation. In this Recommendation, the second unassigned bit is defined to indicate it is an interworking NSH and marked as "IW" in Figure A.2. The flag field "IW" set to "1" means that it is an interworking NSH. The flag field "IW" set to "0" means that it is a general/normal NSH.

There are four fixed length context headers in NSH defined in [IETF RFC 8300]. One or more of them can be used to support the service chain in one SFC domain interworking with another one in other SFC domains. In this Recommendation, *Fixed Length Context Header1* is defined to indicate the interworking flow identifier and address of the SFC interworking function (the address to indicate the flow/packet to be transported back to the SFC interworking function after interworking SFC is completed), and *Fixed Length Context Header2* is defined to carry metadata related to a specific service (e.g., security service, payment service) in Figure A.2.

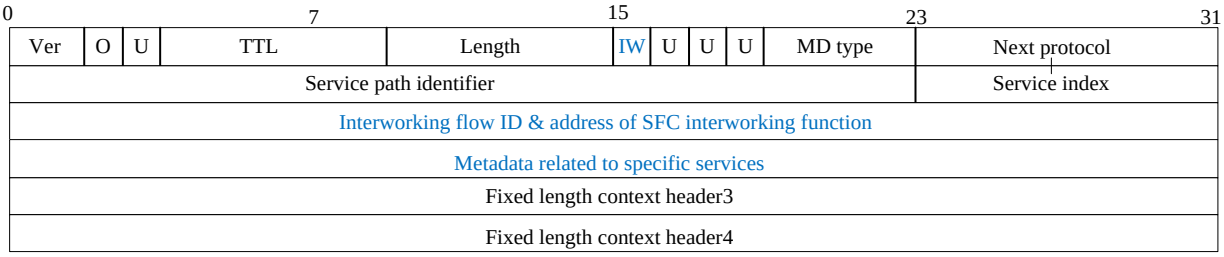


Figure A.2 – NSH extensions to support the service chain in one SFC domain interworking with another one in other SFC domains

A.2 NSH extensions to support customized security protection for data services based on data labelling

Figure A.3 shows the extension of NSH with MD Type=0x1 defined in [IETF RFC 8300] in order to support customized data service security based on data labelling. The data labelling scheme is defined in Annex B of this Recommendation.

NOTE – The extension as below is also applied to NSH with MD Type=0x2 i.e., variable length context headers.

- There are five unassigned bits marked as "U" in the NSH base header. One of them can be used to indicate whether this NSH is a general NSH as defined in [IETF RFC 8300] or extended NSH as defined in this Recommendation. For this Recommendation, the third unassigned bit marked as "U" in the NSH base header is chosen and marked as *DL*. The flag field "DL" set to "1" means that it is an extended NSH used for data service security. The flag field "DL" set to "0" means that it is general/normal NSH.
- There are four fixed length context headers in NSH defined in [IETF RFC 8300]. One or more of them can be extended to support customized data service security based on data labelling. In this Recommendation, *Fixed Length Context Header3* is chosen and used to carry metadata related to data labels.

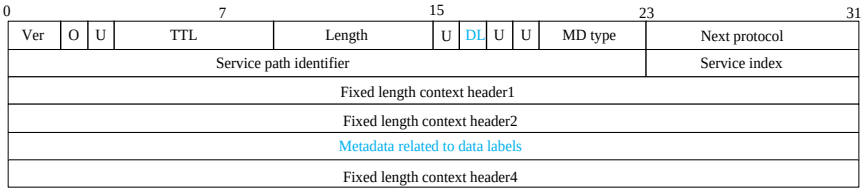


Figure A.3 – NSH extension to support customized data service security

NSH metadata related to data labels (referred to as *Label-NSH* in this Recommendation) is shown in Figure A.4

- Data label: Security level in transit (6 bits): for example, 0x00: security-level-transit-0; 0x01: security-level-transit-1; 0x05: security-level-transit-5;
- Data segment label (22 bits):
  - Label of segment industry (10 bits): for example, 0x00: segment-industry-0; 0x01: segment-industry-1(health); 0x06: segment-industry-6(energy/utilities); 0x07: sub-segment-industry-1(Gas); 0x08: sub-segment-industry-2(Water);
  - Label of segment technology (4 bits): for example, 0x00: segment-technology-0; 0x01: segment-technology-1;
  - Label of segment geography (8 bits): for example, 0x00: segment-geography-0; 0x07: segment-geography-7(e.g., China); 0x09: segment- geography-9(e.g., German);

- Reserved for future (4 bits): undefined

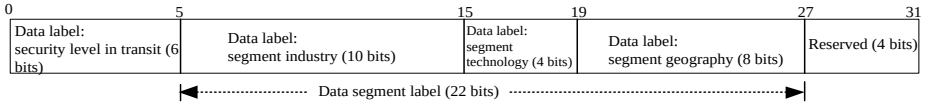


Figure A.4 – Structure of Label-NSH in NSH metadata

A.3 NSH extensions to support tracing network attacks to their sources in SFC overlay network with high performance

Figure A.5 shows the extension of NSH with MD Type=0x1 in order to support tracing network attacks to their sources in SFC overlay network with high performance.

NOTE – The extension as below is also applied to NSH with MD Type=0x2 (i.e., variable length context headers).

- There are five unassigned bits marked as "U" in NSH base header. One of them can be used to indicate whether this NSH is a general NSH defined in [IETF RFC 8300] or extended NSH defined in this Recommendation. For this Recommendation, the fifth unassigned bit marked as "U" in NSH base header is chosen and marked as "TR". The flag field "DL" set to "1" means that it is an extended NSH used to trace network attacks to their sources.
- There are four fixed length context headers in NSH defined in [IETF RFC 8300]. One of them can be extended to support tracing sources of network attacks. In this Recommendation, the first 16 bits of *Fixed Length Context Header4* in Figure A.5 are chosen and used to carry metadata which enables tracing sources of network attacks.
  - It is assumed that NSH information is secured by the outer transportation encapsulation network protocols, such as MPLS, VXLAN/VXLAN-GPE, GRE or IP in IP [IETF RFC 8300]. If the outer transportation encapsulation network protocols could not provide secure protection for NSH information during transportation, NSH with MD Type = 0x2 (i.e., variable length context headers) should be extended to enable confidentiality and integrity protection for NSH information. In this way, four or more variable context headers in the NSH with MD Type = 0x2 will be used to enable security protection for NSH information during transportation. However, the flow/packet fragmentation may occur since the length of encrypted texts or hash value or a digital signature is longer than 128 bits.

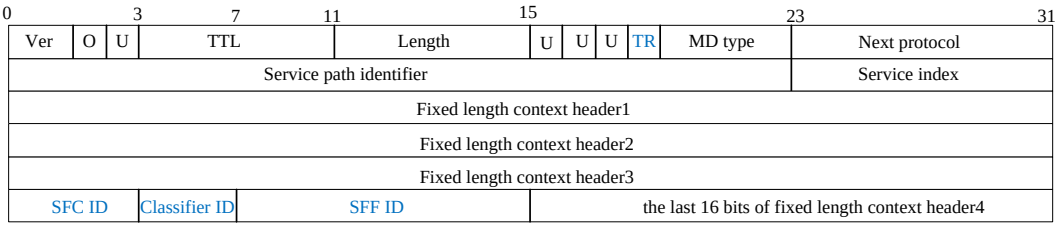


Figure A.5 – Extension of NSH to support tracing network attacks to their sources

Extended NSH metadata related to tracing sources of network attacks in Figure A.5 is described below:

- SFC ID (4 bits): the identifier of the SFC domain which supports tracing network attacks to their sources across SFC domains.
- Classifier ID (4 bits): the identifier of the classifier defined in [IETF RFC 7665] in the specified SFC domain; it is possible that several SFC domains cooperate with each other to complete a transaction or to provide services to end users.

- SFF ID (8 bits): the identifier of SFF defined in [IETF RFC 7665] in the specified SFC domain.

Annex B

Data labelling schemes

(This annex forms an integral part of this Recommendation.)

This annex defines the data labelling schemes which are used for data classification based on data features (e.g., data security, data format, data segment, etc.).

B.1 Data labelling schemes

Data labelling schemes are defined as in Figure B.1.

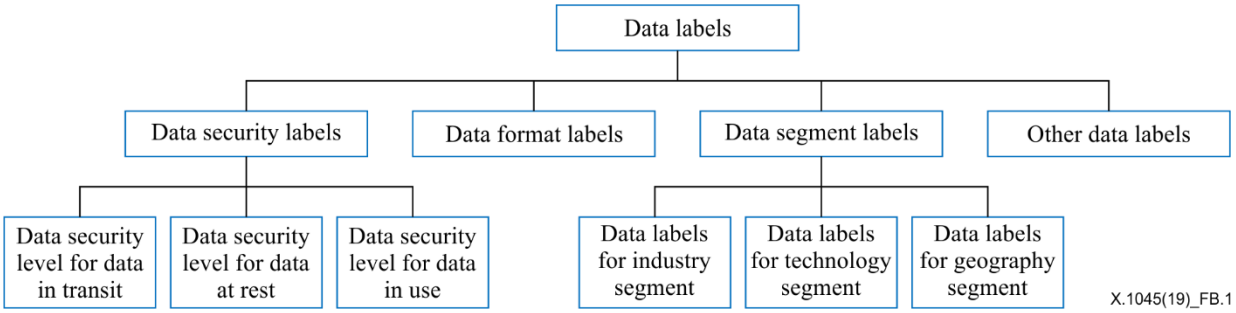


Figure B.1 – Data labelling schemes

The data labels, shown in Figure B.1, are defined below:

Data security labels: store information about data security classification and about which security mechanisms should be applied to data in order to meet the different security requirements of data services. Different data security labels can be defined according to specific security requirements of data services. In this Recommendation the following data security labels are used for example for data in transit, in use, and at rest separately. Data security level labels for *data in transit* are listed below as an example:

- security-level-transit-0: no integrity or confidentiality protection
- security-level-transit-1: integrity protection
- security-level-transit-2: confidentiality protection
- security-level-transit-3: both integrity and confidentiality protection
- security-level-transit-4: both integrity and confidentiality protection and security algorithms with key length of 256 bits or 512 bits

The existing data labelling schemes, such as for example label security [b-LabelSEC], are defined for security protection only for data in use or data at rest. However, security protection for data transportation from data collecting point to a datacentre, data transportation from a datacentre's network perimeter to data storage device and data transportation across geo-distributed datacentres is also a very important complementary security for datacentres which adopt existing database systems. In order to support secure transportation for data with existing data labelling schemes, these data labels for data in use or data at rest can be converted into data labels for data in transit in this Recommendation. For example, the data label "highly sensitive" defined by existing label security for data in use can be converted into the data label "security-level-transit-4" as above. These labels are reflected in NSH metadata in Figure A.4 of Annex 2 for secure transportation.

B.2 To generate data labels and add data labels during data moving in and out of a datacentre

Figure B.2 shows how to classify data, generate data labels (i.e., Label-NSH and Label-P and D), add Label-NSH (in Figure A.4) into NSH metadata and add Label-P and D into payload and database according to data labelling scheme defined in this Recommendation while moving data in a datacentre. In this scenario, there is no data label for the original data which enters into the datacentre.

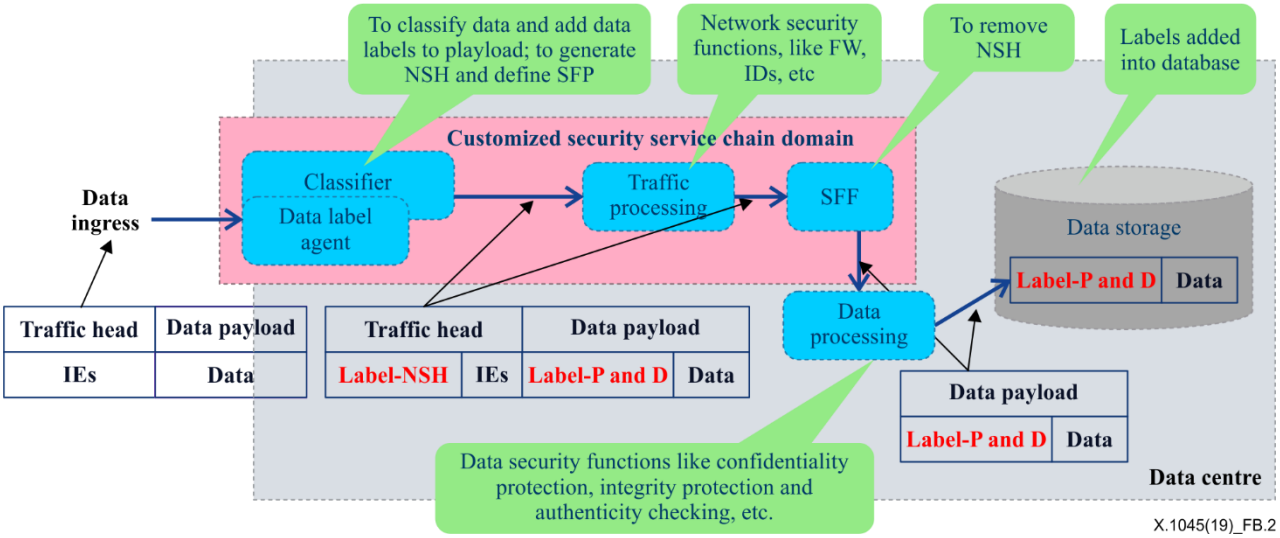


Figure B.2 – Labels added into NSH, payload and database during data moving in datacentre

Figure B.3 shows how to map Label-P and D to Label-NSH and add Label-NSH to NSH metadata during data moving out of a datacentre. In this scenario, there is a data label for the data which moves out of the datacentre.

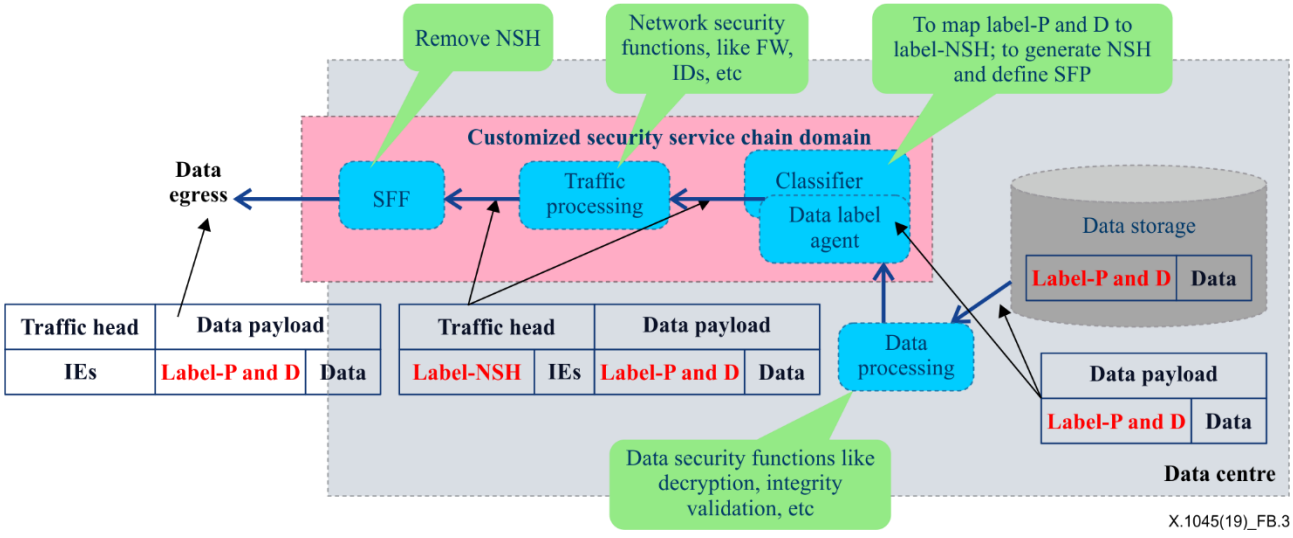


Figure B.3 – Labels added into NSH metadata during data moving out of datacentre

Based on extended NSH with label-NSH, it enables creation of a customized security service chain to provide customized security services for data transportation from a datacentre's network perimeter to data storage devices.

Annex C

Service function chain for special vehicle (SV) speedup

(This annex forms an integral part of this Recommendation.)

This annex is an example to describe how to create a service function chain for special vehicle (SV) speedup.

The ITS SFC domain shown in Figure C.1 can provide some services to citizens such as special vehicle (SV) speedup and bus lane optimization (e.g., private cars can use the bus lane when there is no bus running on the bus-only lane). In this annex, only the service SV speedup and related security protection is described.

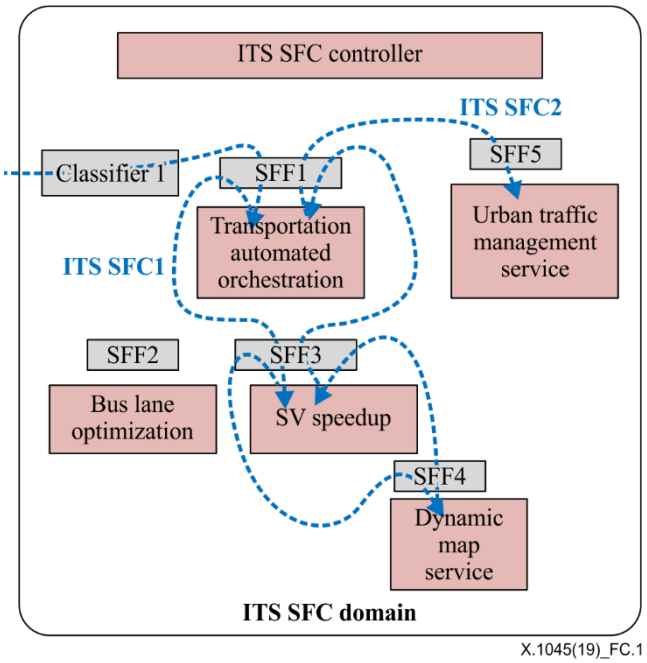


Figure C.1 – Service function chains for special vehicle (SV) speedup

Public safety and disaster rescue services as well as emergency medical services, firefighting and antiterrorism services are more and more important in current society. However, special vehicles such as ambulances, fire engines and police cars also suffer from the traffic congestions which delay the rescue service or public safety services delivery to citizens. So, it is very important to provide a SV speedup service to improve traffic efficiency and safety.

One possible service chain (called ITS SFC1) in Figure C.1 shows how to provide SV speedup service:

Classifier1 -> SFF1 -> Transport automated orchestration -> SFF1 -> SFF3 -> SV speedup -> SFF3 -> SFF4 -> Dynamic map service -> SFF4 -> SFF3 -> SV speedup -> SFF3 -> SFF1 -> Transport automated orchestration.

When a special vehicle is on duty, an ITS station in the special vehicle sends a speedup request to Classifier1. The Classifier1 selects the service forwarding path (SFP) and forwards the request to transport automated orchestration then to SV speedup. SV speedup sends the route request to the dynamic map service. The dynamic map service generates several route options including the identifiers of roads and lanes based on the static map and the status of the road side unit (RSU), e.g., traffic lights and velometer, then responds to the SV speedup. SV speedup selects the best route from the route options according to the criterion (e.g., congestion and short distance) and notifies

the transport automated orchestration to broadcast a speedup pre-emption message (road ID and lane ID to be pre-empted) to other vehicles installed with ITS station. After receiving this pre-emption message, the general vehicles check their current locations and give the way if they are using the pre-emption lane. Then the special vehicle, e.g., a fire engine, can use the pre-emption lane. Once the special vehicle arrives at the destination, it sends a speedup release message to SV speedup to release the pre-emption lane.

Another possible service chain (called ITS SFC2) shown in Figure C.1 supports ITS service to notify the urban traffic management service to control the traffic lights:

Transport automated orchestration -> SFF1 -> SFF5 -> Urban traffic management service.

Transport automated orchestration checks congestion status in this route continually. If the congestion is high, it sends a speedup RSU command to the urban traffic management service to control the traffic lights to be on green or reduce the red light time to reduce congestion in the route.

In order to provide special vehicle speedup service, ITS SFC1 optionally together ITS SFC2 in Figure C.1 will be created.

It is possible that the urban traffic management service could be deployed in a different SFC domain managed by the city traffic police. So, there may be an interworking between ITS SFC domain and another SFC domain which manages the urban traffic management service. In order to simplify the description, it is assumed that the urban traffic management service is deployed in an ITS SFC domain.

Bibliography

[b-CMU Tracing] H.F. Lipson (2002), *Tracking and Tracing Cyber-Attacks: Technical Challenges and Global Policy Issues*, Special Report, CMU/SEI-2002-SR-009, Nov.

[b-FIS 2010] J. Carapinha, P. Feil, P. Weissmann, S. E. Thorsteinsson, C. Etemoglu, O. Ingorsson, S. Ciftci, and M. Melo (2010), *Network Virtualization – Opportunities and Challenges for Operators, in Future Internet* – FIS 2010. Springer Berlin Heidelberg, pp. 138–147.

[b-ICIN] Hu Z., Wang M., Yan X., Yin Y. and Luo Z. (2015), *A Comprehensive Security Architecture for SDN*, the 18th International Conference on Intelligence in Next Generation Networks (ICIN), IEEE, pp. 30-37.

[b-IEEE IP Traceback] A. C. Snoeren, et al., *Single-Packet IP Traceback*, IEEE/ACM Transactions on Networking, vol. 10, no. 6, pp. 721-734.

[b-IWCMC] Hu Z. and Yin Y. (2017), *A Framework for Security on Demand*, the 13th International Conference on Wireless Communications and Mobile Computing (IWCMC), IEEE, pp. 378-383.

[b-LabelSEC] Oracle Label Security, March 2018 http://www.oracle.com/technetwork/wp-dbsec-ols-201702-3634252.pdf

[b-ONF OpenFlow] OpenFlow Switch Specification Version 1.4.0, Open Networking Foundation. www.opennetworking.org/sdn-resources/technical-library

[b-ONF SDN] *Software-Defined Networking (SDN) Definition*, Open Networking Foundation.

[b-SIGCOMM] D. A. Joseph, A. Tavakoli, and I. Stoica (2008), *A policy-aware switching layer for data centers*, in Proceedings of the ACM SIGCOMM 2008 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, Seattle, WA, USA. ACM, pp. 51-62.

[b-Threat Report] McAfee Labs Threats Report. March 2018. https://www.mcafee.com/enterprise/en-us/assets/reports/rp-quarterly-threats-mar-2018.pdf

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series D	Tariff and accounting principles and international telecommunication/ICT economic and policy issues
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	Transmission systems and media, digital systems and networks
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Environment and ICTs, climate change, e-waste, energy efficiency; construction, installation and protection of cables and other elements of outside plant
Series M	Telecommunication management, including TMN and network maintenance
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling, and associated measurements and tests
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	**Data networks, open system communications and security**
Series Y	Global information infrastructure, Internet protocol aspects, next-generation networks, Internet of Things and smart cities
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2020