

Network Working Group
Internet-Draft
Intended status: Experimental
Expires: December 28, 2015

A. Pastor
D. Lopez
Telefonica I+D
K. Wang
X. Zhuang
M. Qi
China Mobile
M. Zarny
Goldman Sachs
S. Majee
F5 Networks
N. Leymann
Deutsche Telekom
L. Dunbar
Huawei
M. Georgiades
PrimeTel
June 26, 2015

Use Cases and Requirements for an Interface to Network Security
Functions
draft-pastor-i2nsf-merged-use-cases-00

Abstract

This document describes use cases and requirements for a common interface to Network Security Functions (NSF). It considers several use cases, organized in two basic scenarios. In the access network scenario, mobile and residential users access NSF capabilities using their network service provider infrastructure. In the data center scenario customers manage NSFs hosted in the data center infrastructure.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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

Internet-Draft Use Cases and Requirements for I2NSF

June 2015

This Internet-Draft will expire on December 28, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

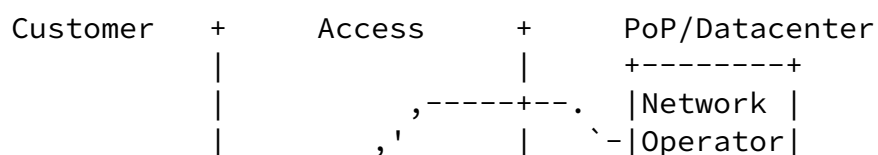
1.	Introduction	4
2.	Requirements Language	5
3.	Terminology	5
4.	General Use Cases	5
4.1.	Instantiation and Configuration of NSFs	6
4.2.	Updating of NSFs	6
4.3.	Collecting the Status of NSFs	6
4.4.	Validation of NSFs	7
5.	Access Network Scenario	7
5.1.	vNSF Deployment	7
5.2.	vNSF Customer Provisioning	7
6.	Cloud Datacenter Scenario	8
6.1.	On-Demand Virtual Firewall Deployment	8
6.2.	Firewall Policy Deployment Automation	9
6.2.1.	Client-Specific Security Policy in Cloud VPNs	9
7.	Considerations on Policy and Configuration	10
7.1.	Translating Policies into NSF Capabilities	11
8.	Key Requirements	12
9.	Security Considerations	13
10.	IANA Considerations	13
11.	References	14
11.1.	Normative References	14
11.2.	Informative References	14
	Authors' Addresses	14

1. Introduction

Not only enterprise customers, but also residential and mobile ones are becoming more and more aware of the need for network security, just to find that security services are hard to operate and become expensive in the case of reasonably sophisticated ones. This general trend has caused numerous operators and security vendors to start to leverage on cloud-based models to deliver security solutions. In particular, the methods around Network Function Virtualization (NFV) are meant to facilitate the elastic deployment of software images providing the network services, and require the management of various resources by customers, who may not own or physically host those network functions.

There are numerous benefits by defining such interfaces. Operators could provide more flexible and customized security services for specific users and this would provide more efficient and secure protection to each user.

This document analyzes the use cases for the provisioning, operation and management of Network Security Functions (NSF) in the access network environment, and cloud-based services as shown in the following figure.



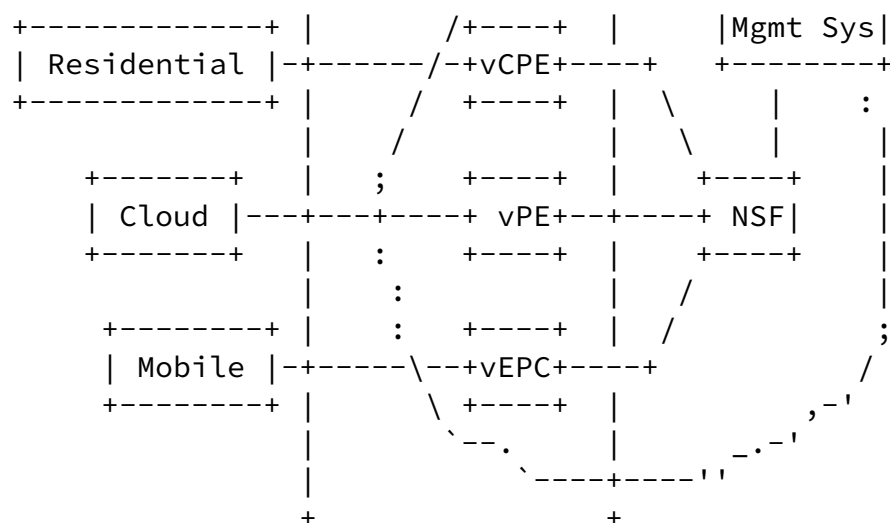


Figure 1: NSF and actors

2. Requirements Language

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

In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [RFC-2119](#) significance.

3. Terminology

Network Security Function (NSF): A functional block within a network infrastructure to ensure integrity, confidentiality and availability of network communications, to detect unwanted activity, and to deter and block this unwanted activity or at least mitigate its effects on the network

vNSF: Virtual Network Security Function: A network security function that runs as a software image on a virtualized infrastructure, and can be requested by one domain but may be owned or managed by another

domain.

NSFs considered in this draft include virtualized and non-virtualized NSFs.

4. General Use Cases

User request security services through specific clients (a customer app, the NSP BSS/OSS or management platform...) and the appropriate NSP network entity will invoke the (v)NSFs according to the user service request. We will call this network entity the security controller. The interaction between the entities discussed above (client, security controller, NSF) is shown in the following diagram:

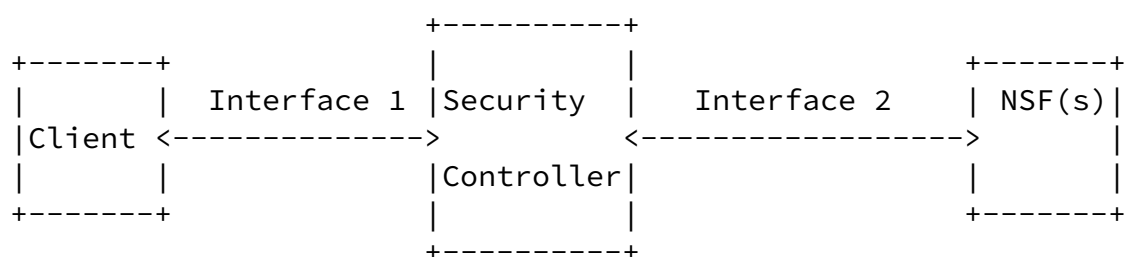


Figure 2: Interaction between Entities

Interface 1 is used for receiving security requirements from client

and translating them into commands that NSFs can understand and execute. Moreover, it is also responsible for giving feedback of the NSF security statistics to client. Interface 2 is used for interacting with NSFs according to commands, and collect status information about NSFs.

4.1. Instantiation and Configuration of NSFs

Client sends collected security requirements through Interface 1 to the security controller in the NSP network, which then translates them into a set of security functions. Then the corresponding NSFs are instantiated and configured through Interface 2.

As an example, consider an enterprise user A who wants to prevent a certain kind of traffic from flowing to their network. Such a

requirement is sent from client to security controller through Interface 1. The security controller translates the requirement into a firewall function plus a rules for filtering out TCP and/or UDP data packets. Then it instantiates a firewall NSF through Interface 2. The corresponding filter rules are also configured onto this firewall NSF through Interface 2.

[4.2.](#) Updating of NSFs

A user can direct the client to require the update of security service functions, including adding/deleting a security service function and updating configurations of former security service function.

As an example, consider a user who has instantiated a security service before and decides to enable an additional IDS service. This requirement will be sent to the security controller through Interface 1 and be translated, so the security controller instantiates and configures an IDS NSF through Interface 2.

[4.3.](#) Collecting the Status of NSFs

When users want to get the executing status of security service, they can request the status statistics information of NSFs from the client. The security controller will collect NSF status statistics information through Interface 2, consolidate them, and give feedback to client through Interface 1. This interface can be used to collect not only individual service information, but also aggregated data suitable for tasks like infrastructure security assessment.

[4.4.](#) Validation of NSFs

Customers may require to validate NSF availability, provenance, and its correct execution. This validation process, especially relevant for vNSFs, includes at least:

- o Integrity of the NSF. Ensure that the NSF is not manipulated.

- o Isolation. The execution of the NSF is self-contained for privacy requirements in multi-tenancy scenarios.

In order to achieve this the security controller has to collect security measurements and share them with an independent and trusted third party, allowing the user to attest the NSF by using Interface 1 and the information of the trusted third party.

[5.](#) Access Network Scenario

This scenario describes use cases for users (enterprise user, network administrator, residential user...) that request and manage security services hosted in the network service provider (NSP) infrastructure. Given that NSP customers are essentially users of their access networks, the scenario is essentially associated with their characteristics, as well as with the use of vNSFs.

The Virtual CPE described in [[NFVUC](#)] use cases #5 and #7 cover the model of virtualization for mobile and residential access, where the operator may offload security services from the customer local environment (or even the terminal) to the operator infrastructure supporting the access network.

These use cases defines the operator interaction with vNSFs through automated interfaces, typically by B2B communications performed by the operator management systems (OSS/BSS).

[5.1.](#) vNSF Deployment

The deployment process consists of instantiating a NSF on a Virtualization Infrastructure (NFVI), within the NSP administrative domain(s) or with other external domain(s). This is a required step before a customer can subscribe to a security service supported in the vNSF.

[5.2.](#) vNSF Customer Provisioning

Once a vNSF is deployed, any customer can subscribe to it. The provisioning lifecycle includes:

- o Customer enrollment and cancellation of the subscription to a

vNSF.

- o Configuration of the vNSF, based on specific configurations, or derived from common security policies defined by the NSP.
- o Retrieve and list of the vNSF functionalities, extracted from a manifest or a descriptor. The NSP management systems can demand this information to offer detailed information through the commercial channels to the customer.

6. Cloud Datacenter Scenario

In a datacenter, network security mechanisms such as firewalls may need to be added or removed dynamically for a number of reasons. It may be explicitly requested by the user, or triggered by a pre-agreed-upon service level agreement (SLA) between the user and the provider of the service. For example, the service provider may be required to add more firewall capacity within a set timeframe whenever the bandwidth utilization hits a certain threshold for a specified period. This capacity expansion could result in adding new instances of firewalls. Likewise, a service provider may need to provision a new firewall instance in a completely new environment due to a new requirement.

The on-demand, dynamic nature of deployment essentially requires that the network security "devices" be in software or virtual form factors, rather than in a physical appliance form. (This is a provider-side concern. Users of the firewall service are agnostic, as they should, as to whether or not the firewall service is run on a VM or any other form factor. Indeed, they may not even be aware that their traffic traverses firewalls.)

Furthermore, new firewall instances need to be placed in the "right zone" (domain). The issue applies not only to multi-tenant environments where getting the tenant right is of paramount importance but also to environments owned and operated by a single organization with its own service segregation policies. For example, an enterprise may mandate that firewalls serving Internet traffic and business-to-business (B2B) traffic be separate; or that IPS/IDS services for investment banking and non-banking traffic be separate for regulatory reasons.

6.1. On-Demand Virtual Firewall Deployment

A service provider operated cloud data center could serve tens of thousands of clients. Clients' compute servers are typically hosted

on virtual machines (VMs), which could be deployed across different server racks located in different parts of the data center. It is often not technically and/or financially feasible to deploy dedicated physical firewalls to suit each client's myriad security policy requirements. What is needed is the ability to dynamically deploy virtual firewalls for each client's set of servers based on established security policies and underlying network topologies.

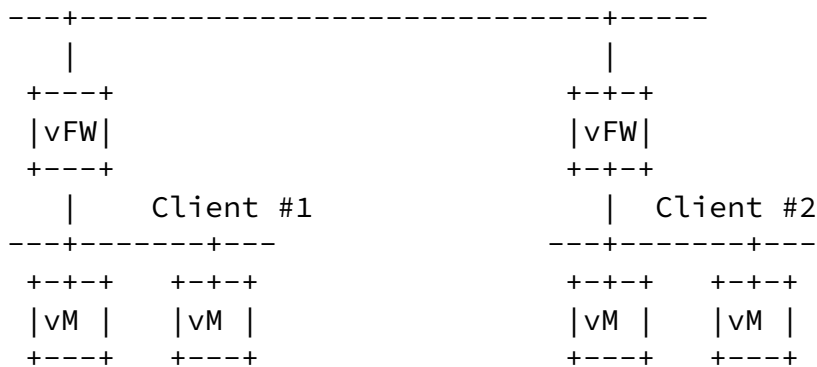


Figure 3: NSF in DataCenter

6.2. Firewall Policy Deployment Automation

Firewall configuration today is a highly complex process that involves consulting established security policies, translating those policies into firewall rules, further translating those rules into vendor-specific configuration sets, identifying all the firewalls, and pushing configurations to those firewalls.

This is often a time consuming, complex and error-prone process even within a single organization/enterprise framework. It becomes far more complex in provider-owned cloud networks that serve myriad customers.

Automation can help address many of these issues. Automation works best when it can leverage a common set of standards that will work across multiple entities.

6.2.1. Client-Specific Security Policy in Cloud VPNs

Clients of service provider operated cloud data centers need not only secure virtual private networks (VPNs) but also virtual security functions that enforce the clients' security policies. The security policies may govern communications within the clients' own virtual networks and those with external networks. For example, VPN service providers may need to provide firewall and other security services to

their VPN clients. Today, it is generally not possible for clients to dynamically view, much less change, what, where and how security

policies are implemented on their provider-operated clouds. Indeed, no standards-based framework that allows clients to retrieve/manage security policies in a consistent manner across different providers exists.

7. Considerations on Policy and Configuration

NSF configurations can vary from simple rules (i.e. block a DDoS attack) to very complex configuration (i.e. define a user firewall rules per application, protocol, source and destination port and address). The possibility of using configuration templates per control and management type is a common option as well.

A NSP can push security policies using complex configurations in their managed vNSF through its management system. The open Control and management interface has to accommodate this application-driven behavior.

Computer-savvy customers may pursue a similar application-driven configuration through the open Control and management interface, but standard residential and mobile customers may prefer to use the definition of security policies in the form of close-to-natural-language sentences with high-level directives or a guide configuration process. The representation for these policies will be of the form:

```
+-----+ +-----+ +-----+ +-----+
|Subject| + |Action| + |Object| + |Field_type = Value|
+-----+ +-----+ +-----+ +-----+
```

Figure 4: High-Level Security Policy Format

Subject indicates the customer or device in the access.

Action can include a variety of intent-based actions: check, redirect, allow, block, record, inspect...

Object can be optional and specifies the nature of the action. The default is all the customer traffic, but others possible values are connections and connections attempts.

Field_type allows to create fine-grained policies, including destinations list (i.e. IPs, domains), content types (i.e. files, emails), windows of time (i.e. weekend), protocol or network service (i.e. HTTP).

An example of a customer policy is:

"My son is allowed to access Facebook from 18:30 to 20:00"

[7.1.](#) Translating Policies into NSF Capabilities

Policies expressed in the above model are suitable for what we depicted as Interface 1 in Figure 2. In order to allow the security controller to deal with the different NSFs an intermediate representation used for expressing specific configurations in a device-independent format is required. For this purpose, the definition of a set of security capabilities provides a means for categorizing the actions performed by network security functions. An initial, high-level set of such capabilities consists of:

- o Identity Management: Includes all services related with identity, authentication and key management. Some examples are:
 - * AAA (Authentication, Authorization, Accounting) services
 - * Remote identity management
 - * Secure key management
- o Traffic Inspection: A common use case for customers accessing the Internet or additional services through it is security supervision. Control and Management interfaces will allow the configuration of the vNSF inspection features: signatures updates, behavioral parameters or type of traffic to supervise. Some examples are:
 - * IDS/IPS (Intrusion Detection System/Intrusion Prevention System

- * Deep packet inspection
- * Data leakage protection
- o Traffic Manipulation: A more intrusive use case of NSF includes the capacity of manipulate the client traffic. Control and Management interfaces will allow the configuration of the NSF manipulation features, such as redirect and block rules. Some examples are:
 - * Redirect traffic, as in the case of captive portals
 - * Block traffic: Firewalls, intrusion prevention system, DDOS/ Anti-DOS (Distributed Denial-of-Service/Anti-Denial-of-Service)

- * Encrypt traffic: VPN services that encapsulate and encrypt the user traffic. A SSL VPN is a representative example.
- o Impersonation: Some NSFs can impersonate a customer service or Internet service to provide security functions. Control and Management interfaces will allow the configuration of the service to impersonate and his behavioral. Some examples are:
 - * Honeypots, impersonating customer services, such as HTTP, NetBios or SSH
 - * Anonymization services, hiding the source identity, as in the case of TOR

Service Chain will allow for more than one of the aforementioned functions to engage in a specific order to a particular flow

[8.](#) Key Requirements

The I2NSF framework should provide a set of standard interfaces that facilitate:

- o Dynamic creation, enablement, disablement, and removal of network security functions;

- o Policy-driven placement of new function instances in the right administrative domain;
- o Attachment of appropriate security and traffic policies to the function instances
- o Management of deployed instances in terms of fault monitoring, utilization monitoring, event logging, inventory, etc.

Moreover, an I2NSF must support different deployment scenarios:

- o Single and multi-tenant environments: The term multi-tenant does not mean just different companies subscribing to a provider's offering. It can for instance cover administrative domains/ departments within a single firm that require different security and traffic policies.
- o Premise-agnostic: Said network security functions may be deployed on premises or off premises of an organization.

The I2NSF framework should provide a standard set of interfaces that enable:

- o Translation of security policies into functional tasks. Security policies may be carried out by one or more security functions. For example, a security policy may be translated into an IDS/IPS policy and a firewall policy for a given application type.
- o Translation of functional tasks into vendor-specific configuration sets. For example, a firewall policy needs to be converted to vendor-specific configurations.
- o Retrieval of information such as configuration, utilization, status, etc. Such information may be used for monitoring, auditing, troubleshooting purposes. The above functionality should be available in single- or multi-tenant environments as well as on-premise or off-premise clouds.

[9.](#) Security Considerations

The relationship between different actors define the security level for the different use cases and must be associated with administrative domains:

- o Closed environments where there is only one administrative network domain. More permissive access controls and lighter validation shall be allowed inside the domain because of the protected environment. Integration with existing identity management systems is also possible.
- o Open environments where some NSFs can be hosted in different administrative domains, and more restrictive security controls are required. The interfaces to the NSFs must use trusted channels. Identity frameworks and federations are common models for authentication and Authorization. Security controllers will be in charge of this functionalities.

Virtualization applied to NSF environment (vNSF) generate several concerns in security, being one of the most relevant the attestation of the vNSF by the clients. A holistic analysis has been done in [\[NFVSEC\]](#).

[10.](#) IANA Considerations

This document requires no IANA actions.

[11.](#) References

Pastor, et al.	Expires December 28, 2015	[Page 13]
----------------	---------------------------	-----------

Internet-Draft	Use Cases and Requirements for I2NSF	June 2015
----------------	--------------------------------------	-----------

[11.1.](#) Normative References

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

[11.2.](#) Informative References

[NFVSEC] "ETSI NFV Group Specification, Network Functions Virtualization (NFV) NFV Security; Problem Statement", http://www.etsi.org/deliver/etsi_gs/NFV-SEC/001_099/001/01.01.01_60/gs_NFV-SEC001v010101p.pdf.

[NFVUC] "ETSI NFV Group Specification, Network Functions Virtualization (NFV) Use Cases", <http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf>.

Authors' Addresses

Antonio Pastor
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid, 28006
Spain

Phone: +34 913 128 778
Email: antonio.pastorperales@telefonica.com

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid, 28006
Spain

Phone: +34 913 129 041
Email: diego.r.lopez@telefonica.com

Ke Wang
China Mobile
32 Xuanwumenxi Ave,Xicheng District
Beijing, 100053
China

Email: wangkeyj@chinamobile.com

Pastor, et al.

Expires December 28, 2015

[Page 14]

Internet-Draft Use Cases and Requirements for I2NSF

June 2015

Xiaojun Zhuang
China Mobile
32 Xuanwumenxi Ave,Xicheng District
Beijing, 100053

China

Email: zhuangxiaojun@chinamobile.com

Minpeng Qi
China Mobile
32 Xuanwumenxi Ave, Xicheng District
Beijing, 100053
China

Email: quiminpeng@chinamobile.com

Myo Zarny
Goldman Sachs
30 Hudson Street
Jersey City, NJ 07302
USA

Email: myo.zarny@gs.com

Sumandra Majee
F5 Networks

Email: lal2ghar@gmail.com

Nic Leymann
Deutsche Telekom

Email: n.leymann@telekom.de

Linda Dunbar
Huawei

Email: linda.dunbar@huawei.com

Michael Georgiades
PrimeTel

Email: michaelg@prime-tel.com

