I2NSF
Internet-Draft
Intended status: Informational
Expires: November 11, 2017

S. Hares
Huawei
D. Lopez
Telefonica I+D
M. Zarny
vArmour
C. Jacquenet
France Telecom
R. Kumar
Juniper Networks
J. Jeong
Sungkyunkwan University

May 10, 2017

I2NSF Problem Statement and Use cases
draft-ietf-i2nsf-problem-and-use-cases-16

Abstract

This document is the problem statement for Interface to Network Security Functions (I2NSF) as well as some companion use cases.

Status of This Memo

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

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

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

This Internet-Draft will expire on November 11, 2017.

Copyright Notice

Table of Contents

1. Introduction

This document is the problem statement for Interface to Network Security Functions (I2NSF) as well as some I2NSF use cases. A summary of the state of the art in the industry and IETF which is relevant to I2NSF work is documented in [I-D.ietf-i2nsf-gap-analysis].

The growing challenges and complexity in maintaining a secure infrastructure, complying with regulatory requirements, and controlling costs are enticing enterprises into consuming network security functions hosted by service providers. The hosted security service is especially attractive to small and medium size enterprises who suffer from a lack of security experts to continuously monitor networks, acquire new skills and propose immediate mitigations to ever increasing sets of security attacks.

According to [Gartner-2013], the demand for hosted (or cloud-based) security services is growing. Small and medium-sized businesses (SMBs) are increasingly adopting cloud-based security services to replace on-premises security tools, while larger enterprises are deploying a mix of traditional and cloud-based security services.

To meet the demand, more and more service providers are providing hosted security solutions to deliver cost-effective managed security services to enterprise customers. The hosted security services are primarily targeted at enterprises (especially small/medium ones), but could also be provided to any kind of mass-market customer. As a result, the Network Security Functions (NSFs) are provided and consumed in a large variety of environments. Users of NSFs may consume network security services hosted by one or more providers, which may be their own enterprise, service providers, or a combination of both.

This document also briefly describes the following use cases summarized by [I-D.pastor-i2nsf-merged-use-cases]:

- [I-D.pastor-i2nsf-access-usecases] (I2NSF-Access),
- [I-D.zarny-i2nsf-data-center-use-cases](I2NSF-DC), and
- [I-D.qi-i2nsf-access-network-usecase] (I2NSF-Mobile).

2. Terminology

AAA: Authentication, Authorization, and Account [RFC2904].

ACL: Access Control List

Bespoke security management: Security management which is made to fit a particular customer.

DC: Data Center

vEPC virtual 3GPP Evolved Packet Core [EPC-3GPP]

FW: Firewall

IDS: Intrusion Detection System

IPS: Intrusion Protection System

I2NSF: Interface to Network Security Functions

NSF: Network Security Function. An NSF is a function that used to ensure integrity, confidentiality, or availability of network communication, to detect unwanted network activity, or to block or at least mitigate the effects of unwanted activity.

Flow-based NSF: An NSF which inspects network flows according to a security policy. Flow-based security also means that packets are inspected in the order they are received, and without altering packets due to the inspection process (e.g., MAC rewrites, TTL decrement action, or NAT inspection or changes). (Note: Some existing firewalls store packets and look at the packets in logical order which is not order these are received in time. This document restricts flow-based NSF to this definition.)

Security service provider: A provider of security services to the customers (end-users or enterprises) using NSF equipment purchased from vendors or created by the service provider.

SDN: Software Defined Networking. (See [RFC7426]) for architectural and terminology or [RFC7149] for service provider view).

Virtual NSF: An NSF which is deployed as a distributed virtual resource.

VPN: Virtual Private Networks

3. Problem Space

The following sub-sections describe the problems and challenges facing customers and security service providers when some or all of the security functions are no longer physically hosted by the customer’s administrative domain.

Security service providers can be internal or external to the company. For example, an internal IT Security group within a large enterprise could act as a security service provider for the enterprise. In contrast, an enterprise could outsource all security services to an external security service provider. In this document, the security service provider function, whether it is internal or external, will be denoted as "service provider".

The "Customer-Provider" relationship may be between any two parties. The parties can be in different organization or different domains of the same organization. Contractual agreements may be required in such contexts to formally document the customer’s security requirements and the provider’s guarantees to fulfill those requirements. Such agreements may detail protection levels, escalation procedures, alarms reporting, etc. There is currently no standard mechanism to capture those requirements.

A service provider may be a customer of another service provider.

It is the objective of the I2NSF work to address these problems and challenges.

3.1. Challenges Facing Security Service Providers

3.1.1. Diverse Types of Security Functions

There are many types of NSFs. NSFs by different vendors can have different features and have different interfaces. NSFs can be deployed in multiple locations in a given network, and perhaps have different roles.

Below are a few examples of security functions and locations or contexts in which they are often deployed:

External Intrusion and Attack Protection: Examples of this function are firewall/ACL authentication, IPS, IDS, and endpoint protection.

Security Functions in a Demilitarized Zone (DMZ): Examples of this function are firewall/ACLs, IDS/IPS, one or all of AAA services, NAT, forwarding proxies, and application filtering. These

functions may be physically on-premise in a server provider's network at the DMZ spots or located in a "virtual" DMZ.

Centralized or Distributed security functions: The security functions could be deployed in a centralized fashion for ease of management and network design or in a distributed fashion for scaled requirement. No matter how a security function is deployed and provisioned, it is desirable to have same interface to provision security policies; otherwise it makes the job of security administration more complex, requiring knowledge of firewall and network design.

Internal Security Analysis and Reporting: Examples of this function are security logs, event correlation, and forensic analysis.

Internal Data and Content Protection: Examples of this function are encryption, authorization, and public/private key management for internal database.

Security gateways and VPN concentrators: Examples of these functions are; IPsec gateways, secure VPN concentrators that handle bridging secure VPNs, and secure VPN controllers for data flows.

Given the diversity of security functions, the contexts in which these functions can be deployed, and the constant evolution of these functions, standardizing all aspects of security functions is challenging, and most probably not feasible. Fortunately, it is not necessary to standardize all aspects. For example, from an I2NSF perspective, there is no need to standardize how every firewall's filtering is created or applied. Some features in a specific vendor's filtering may be unique to the vendor's product so it is not necessary to standardize these features.

What is needed is a standardized interface to control and monitor the rule sets that NSFs use to treat packets traversing through these NSFs. Thus standardizing interfaces will provide an impetus for standardizing established security functions.

I2NSF may specify some filters, but these filters will be linked to specific common functionality developed by I2NSF in information models or data models.

3.1.2. Diverse Interfaces to Control and Monitor NSFs

To provide effective and competitive solutions and services, security service providers may need to utilize multiple security functions

from various vendors to enforce the security policies desired by their customers.

Since no widely accepted industry standard security interface to security NSFs exists today, management of NSFs (device and policy provisioning, monitoring, etc.) tends to be custom-made security management offered by product vendors. As a result, automation of such services, if it exists at all, is also custom-made. Thus, even in the traditional way of deploying security features, there is a gap to coordinate among implementations from distinct vendors. This is the main reason why mono-vendor security functions are often deployed and enabled in a particular network segment.

A challenge for monitoring prior to mitigation of a security intrusion is that an NSF cannot monitor what it cannot view. For example, enabling a security function to mitigate an intrusion (e.g., firewall [I-D.ietf-opsawg-firewalls]) must include a mechanism to provide monitoring feedback in order to determine the intrusion has been stopped. Therefore, it is necessary to have a mechanism to monitor and provide execution status of NSFs to security and compliance management tools. There exist such mechanisms in vendor-specific network security interfaces for forensics and troubleshooting, but an industry standard interface could provide monitoring across a variety of NSFs.

3.1.3. More Distributed NSFs and vNSFs

The security functions which are invoked to enforce a security policy can be located in different equipment and network locations.

The European Telecommunications Standards Institute (ETSI) Network Functions Virtualization (NFV) initiative [ETSI-NFV] creates new management challenges for security policies to be enforced by distributed virtual network security functions (vNSF).

A vNSF has higher risk of changes to the state of network connection, interfaces, or traffic as their hosting Virtual Machines (VMs) are being created, moved, or decommissioned.

3.1.4. More Demand to Control NSFs Dynamically

In the advent of Software-Defined Networking (SDN)(see [I-D.jeong-i2nsf-sdn-security-services]), more clients, applications or application controllers need to dynamically update their security policies that are enforced by NSFs. The security service providers have to dynamically update their decision-making process (e.g., in terms of NSF resource allocation and invocation) upon receiving security-related requests from their clients.

3.1.5. Demand for Multi-Tenancy to Control and Monitor NSFs

Service providers may need to deploy several NSF controllers to control and monitor the NSFs, especially when NSFs become distributed and virtualized.

3.1.6. Lack of Characterization of NSFs and Capability Exchange

To offer effective security services, service providers need to activate various security functions in NSFs or vNSFs manufactured by multiple vendors. Even within one product category (e.g., firewall), security functions provided by different vendors can have different features and capabilities. For example, filters that can be designed and activated by a firewall may or may not support IPv6 depending on the firewall technology.

The service provider’s management system (or controller) needs a way to retrieve the capabilities of service functions by different vendors so that it could build an effective security solution. These service function capabilities can be documented in a static manner (e.g., a file) or via an interface which accesses a repository of security function capabilities which the NSF vendors dynamically update.

A dynamic capability registration is useful for automation because security functions may be subject to software and hardware updates. These updates may have implications on the policies enforced by the NSFs.

Today, there is no standard method for vendors to describe the capabilities of their security functions. Without a common technical framework to describe the capabilities of security functions, service providers cannot automate the process of selecting NSFs by different vendors to accommodate customer’s security requirements.

The I2NSF work will focus on developing a standard method to describe capabilities of security functions.

3.1.7. Lack of Mechanism for NSFs to Utilize External Profiles

Many security functions depend on signature files or profiles (e.g., IPS/IDS signatures, DDos Open Threat Signaling (DOTS) filters). Different policies might need different signatures or profiles. Today, black list databases can be a beneficial strategy for all parties involved (except the attackers), but in the future there might be open Source-provided signature/profiles distributed as part of IDS systems (e.g., by Snort, Suricata, Bro and Kismet).

There is a need to have a standard envelope (i.e., a message format) to allow NSFs to use external profiles.

3.1.8. Lack of Mechanisms to Accept External Alerts to Trigger Automatic Rule and Configuration Changes

NSF can ask the I2NSF security controller to alter specific rules and/or configurations. For example, a Distributed Denial of Service (DDoS) alert could trigger a change to the routing system to send traffic to a traffic scrubbing service to mitigate the DDoS.

The DDoS protection has the following two parts: a) the configuration of signaling of open threats and b) DDoS mitigation. DOTS controller manages the signaling part of DDoS. I2NSF controller(s) would control any changes to affected policies (e.g., forwarding and routing, filtering, etc.). By monitoring the network alerts regarding DDoS attacks (e.g. from DOTS servers or clients), the I2NSF controller(s) can feed an alerts analytics engine that could recognize attacks so the I2NSF can enforce the appropriate policies.

DDoS mitigation is enhanced if the provider’s network security controller can monitor, analyze, and investigate the abnormal events and provide information to the customer or change the network configuration automatically.

[I-D.zhou-i2nsf-capability-interface-monitoring] provides details on how monitoring aspects of the flow-based Network Security Functions (NSFs) can use the I2NSF interfaces to receive traffic reports and enforce appropriate policies.

3.1.9. Lack of Mechanism for Dynamic Key Distribution to NSFs

There is a need for a controller to create, manage, and distribute various keys to distributed NSFs. While there are many key management methods and cryptographic suites (e.g., encryption algorithms, key derivation functions, etc.) and other functions, there is a lack of a standard interface to provision and manage security associations.

The keys may be used for message authentication and integrity in order to protect data flows. In addition, keys may be used to secure the protocols and messages in the core routing infrastructure (see [RFC4948])

As of now there is not much focus on an abstraction for keying information that describes the interface between protocols, operators, and automated key management.

An example of a solution may provide some insight into why the lack of a mechanism is a problem. If a device had an abstract key table maintained by security services, it could use these keys for routing and security devices.

What does this take?

Conceptually, there must be an interface defined for routing/ signaling protocols that can: a) make requests for automated key management when it is being used. and b) notify the protocols when keys become available in the key table. One potential use of such an interface is to manage IPsec security associations on SDN networks.

An abstract key service will work under the following conditions:

1. I2NSF needs to design the key table abstraction, the interface between key management protocols and routing/other protocols, and possibly security protocols at other layers.

2. For each routing/other protocol, I2NSF needs to define the mapping between how the protocol represents key material and the protocol-independent key table abstraction. If several protocols share common mechanisms for authentication (e.g., TCP Authentication Option [RFC5925]), then the same mapping may be used for all usages of that mechanism.

3. Automated key management needs to support both pair-wise keys and group keys via the abstract key service provided by items 1 and 2. I2NSF controllers within the NSF required to exchange data with NSFs may exchange data with individual NSFs using individual pair-wise keys or with a group of NSFs simultaneously using an IP group address secured by a group security key(s).

3.2. Challenges Facing Customers

When customers invoke hosted security services, their security policies may be enforced by a collection of security functions hosted in different domains. Customers may not have the security skills to express sufficiently precise requirements or security policies. Usually, these customers express the expectations of their security requirements or the intent of their security policies. These expectations can be considered customer-level security expectations. Customers may also desire to express guidelines for security management. Examples of such guidelines include:

- Which critical communications are to be preserved during critical events and which hosts will continue services over the network,

- What signaling information is passed to a controller during a Distributed Denial of Service in order to ask for mitigation services (within the scope DOTS working group),

- Reporting of attacks to CERT (within the scope of MILE working group), and

- Managing network connectivity of systems out of compliance (within the scope of SACM working group).

3.2.1. NSFs from Heterogeneous Administrative Domains

Many medium and large enterprises have deployed various on-premises security functions which they want to continue to use. These enterprises want to combine local security functions with remote hosted security functions to achieve more efficient and immediate counter-measures to both Internet-originated attacks and enterprise network-originated attacks.

Some enterprises may only need the hosted security services for their remote branch offices where minimal security infrastructures/ capabilities exist. The security solution will consist of deploying NSFs on customer networks and on service provider networks.

3.2.2. Today’s Control Requests are Vendor Specific

Customers may utilize NSFs provided by multiple service providers. Customers need to express their security requirements, guidelines, and expectations to the service providers. In turn, the service providers must translate this customer information into customer security policies and associated configuration tasks for the set of security functions in their network. Without a standardized interface that provides a clear technical characterization, the service provider faces many challenges:

No standard technical characterization, APIs, or Interface: Even for the most common security services there is no standard technical characterization, APIs, or interface(s). Most security services are accessible only through disparate, proprietary interfaces (e.g., portals or APIs) in whatever format vendors choose to offer. The service provider must process the customer’s input with these widely varying interfaces and differing configuration models for security devices and security policy. Without a standard interface, new innovative security products find a large barrier to entry into the market.

Lack of immediate feedback: Customers may also require a mechanism to easily update/modify their security requirements with immediate effect in the underlying involved NSFs.

Lack of explicit invocation request: While security agreements are in place, security functions may be solicited without requiring an explicit invocation means. Nevertheless, some explicit invocation means may be required to interact with a service function.

Managing by scripts de-jour: The current practices rely upon the use of scripts that generate other scripts which automatically run to upload or download configuration changes, log information and other things. These scripts have to be adjusted each time an implementation from a different vendor technology is enabled by a provider.

To see how standard interfaces could help achieve faster implementation time cycles, let us consider a customer who would like to dynamically allow an encrypted flow with specific port, src/dst addresses or protocol type through the firewall/IPS to enable an encrypted video conferencing call only during the time of the call. With no commonly accepted interface in place, as shown in figure 1, the customer would have to learn about the particular provider's firewall/IPS interface and send the request in the provider's required format.

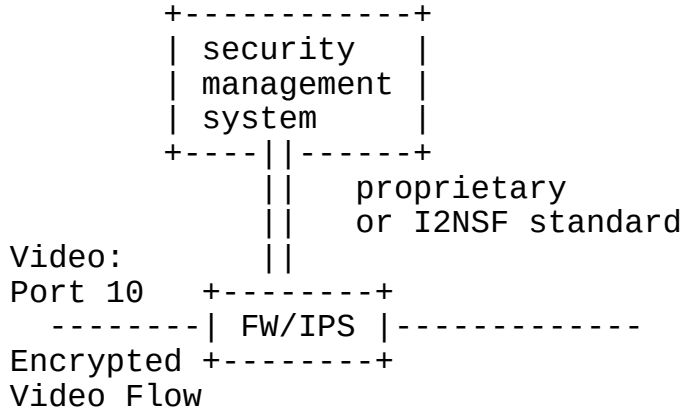

Figure 1: Example of non-standard vs. standard interface

In contrast, as figure 1 shows, if a firewall/IPS interface standard exists the customer would be able to send the request to a security management system and the security management would send it via a I2NSF standard interface. Service providers could now utilize the same standard interface interface to represent firewall/IPS services implemented using products from many vendors.

3.2.3. Difficult for Customer to Monitor the Execution of Desired Policies

How a policy is translated into technology-specific actions is hidden from the customers. However, customers still need ways to monitor the delivered security service that results from the execution of their desired security requirements, guidelines and expectations. Customers want to monitor existing policies to determine such things as: which policies are in effect, how many security attacks are being prevented, and how much bandwidth efficiency does security enforcement cost.

Today, there is no standard way for customers to get these details from the security service which assure the customer that their specified security policies properly enforced by the security functions in the provider domain.

Customers also want this monitoring information from the security system in order to plan for the future using "what-if" scenarios with real data. A tight loop between the data gathered from security systems and the "what-if" scenario planning can reduce the time to design and deploy workable security policies that deal with new threats.

It is the objective of the I2NSF work to provide a standard way to get the information that security service assurance systems need to provide customers an evaluation about the current security systems, and to quickly plan for future security policies using "what-if" scenarios based on today’s information

3.3. Lack of Standard Interface to Inject Feedback to NSF

Today, many security functions in the NSF, such as IPS, IDS, DDoS mitigation and antivirus, depend heavily on the associated profiles. NSF devices can perform more effective protection if these NSF devices have the up-to-date profiles for these functions. Today there is no standard interface to provide these security profiles for the NSF.

As more sophisticated threats arise, protection will depend on enterprises, vendors, and service providers being able to cooperate to develop optimal profiles such as the [CTA]. The standard interface to provide security profiles to the NSF should interwork with the formats which exchange security profiles between organizations.

One objective of the I2NSF work is to provide this type of standard interface to security profiles.

3.4. Lack of Standard Interface for Capability Negotiation

There could be situations when the selected NSFs cannot perform the policies requested by the Security Controller, due to resource constraints. The customer and security service provider should negotiate the appropriate resource constraints before the security service begins. However, unexpected events may happen causing the NSF to exhaust those negotiated resources. At this point, the NSF should inform the security controller that the alloted resources have been exhausted. To support the automatic control in the SDN-era, it is necessary to have a set of messages for proper notification (and a response to that notification) between the Security Controller and the NSFs.

3.5. Difficult to Validate Policies across Multiple Domains

As discussed in the previous four sections, both service providers and customers have need to express policies and profiles, monitor systems, verify security policy has been installed in NSFs within a security domain, and establish limits for services NSFs can safely perform. This sub-section and the next sub-section (section 3.6) examine what happens in two specific network scenarios: a) multi-domain control of security devices hosted on virtual and non-virtual NSFs, and b) software defined networking.

Hosted security service may instantiate NSFs in virtual machines which are sometimes widely distributed in the network and sometimes are combined together in one device to perform a set of tasks for delivering a service. Hosted security services may be connected within a single service provider or via multiple services provider. Ensuring that the security service purchased by the customer adheres to customer policy requires that the central controller(s) for this service monitor and validate this service across multiple networks on NSFs (some of which may be virtual networks on virtual machines). To set up this cross-domain service, the security controller must be able to communicate with NSFs and/or controllers within its domain and across domains to negotiate for the services needed.

Without standard interfaces and security policy data models, the enforcement of a customer-driven security policy remains challenging because of the inherent complexity created by combining the invocation of several vendor-specific security functions into a multi-vendor, heterogeneous environment across multiple domains. Each vendor-specific function may require specific configuration procedures and operational tasks.

Ensuring the consistent enforcement of the policies at various domains is also challenging. Standard data models are likely to contribute to solving that issue.

3.6. Software-Defined Networks

Software-Defined Networks have changed the landscape of data center designs by introducing overlay networks deployed over Top of Rack (ToR) switches that connect to a hypervisor. SDN techniques are meant to improve the flexibility of workload management without affecting applications and how they work. Workload can thus be easily and seamlessly managed across private and public clouds. SDN techniques optimize resource usage and are now being deployed in various networking environments, besides cloud infrastructures. Yet, such SDN conferred agility may raise specific security issues. For example a security administrator must make sure that a security policy can be enforced regardless of the location of the workload, thereby raising concerns about the ability of SDN computation logic to send security policy-provisioning information to the participating NSFs. A second example is workload migration to a public cloud infrastructure which may raise additional security requirements during the migration.

4. Use Cases

Standard interfaces for monitoring and controlling the behavior of NSFs are essential building blocks for security service providers and enterprises to automate the use of different NSFs from multiple vendors by their security management entities. I2NSF may be invoked by any (authorized) client. Examples of authorized clients are upstream applications (controllers), orchestration systems, and security portals.

4.1. Basic Framework

Users request security services through specific clients (e.g., a customer application, the Network Service Providers (NSP) Business Support Systems/Operations Support Systems (BSS/OSS) or management platform) and the appropriate NSP network entity will invoke the (v)NSFs according to the user service request. This network entity is denoted as the security controller in this document. The interaction between the entities discussed above (client, security controller, NSF) is shown in Figure 2:

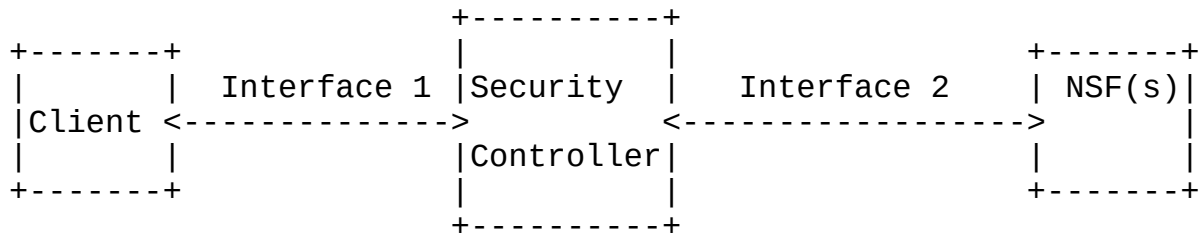

Figure 2: Interaction between Entities

Interface 1 is used for receiving security requirements from a client and translating them into commands that NSFs can understand and execute. The security controller also passes back NSF security reports (e.g., statistics) to the client which the security controller has gathered from NSFs. Interface 2 is used for interacting with NSFs according to commands (e.g., enact/revoke a security policy, or distribute a policy), and collecting status information about NSFs.

Client devices or applications can require the security controller to add, delete or update rules in the security service function for their specific traffic.

When users want to get the executing status of a security service, they can request NSF status from the client. The security controller will collect NSF information through Interface 2, consolidate it, and give feedback to the 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.

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

Integrity of the NSF: Ensuring that the NSF is not compromised;

Isolation: Ensuring the execution of the NSF is self-contained for privacy requirements in multi-tenancy scenarios; and

Provenance of the NSF: Customers may need to be provided with strict guarantees about the origin of the NSF, its status (e.g., available, idle, down, and others), and feedback mechanisms so that a customer may be able to check that a given NSF or set of NSFs properly conform to the the customer’s requirements and subsequent configuration tasks.

In order to achieve this, the security controller may collect security measurements and share them with an independent and trusted third party (via Interface 1) in order to allow for attestation of NSF functions using the third party added information.

This implies that there may be the following two types of clients using interface 1: the end-user and and the trusted independent third party. The I2NSF work may determine that interface 1 creates two sub-interfaces to support these two types of clients.

4.2. Access Networks

This scenario describes use cases for users (e.g., residential user, enterprise user, mobile user, and management system) that request and manage security services hosted in the 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. Figure 3 shows how different types of customer connect through virtual access nodes (vCPE, vPE, and vEPC) to an NSF.

The virtual customer premise equipment (vCPE) described in use case #7 in [NFVUC] requires a model of access virtualization that includes mobile and residential access networks where the operator may offload security services from the customer local environment (e.g., device or terminal) to its own infrastructure.

These use cases define the interaction between the operator and the vNSFs through automated interfaces which support the business communications between customer and provider or between two business entities.

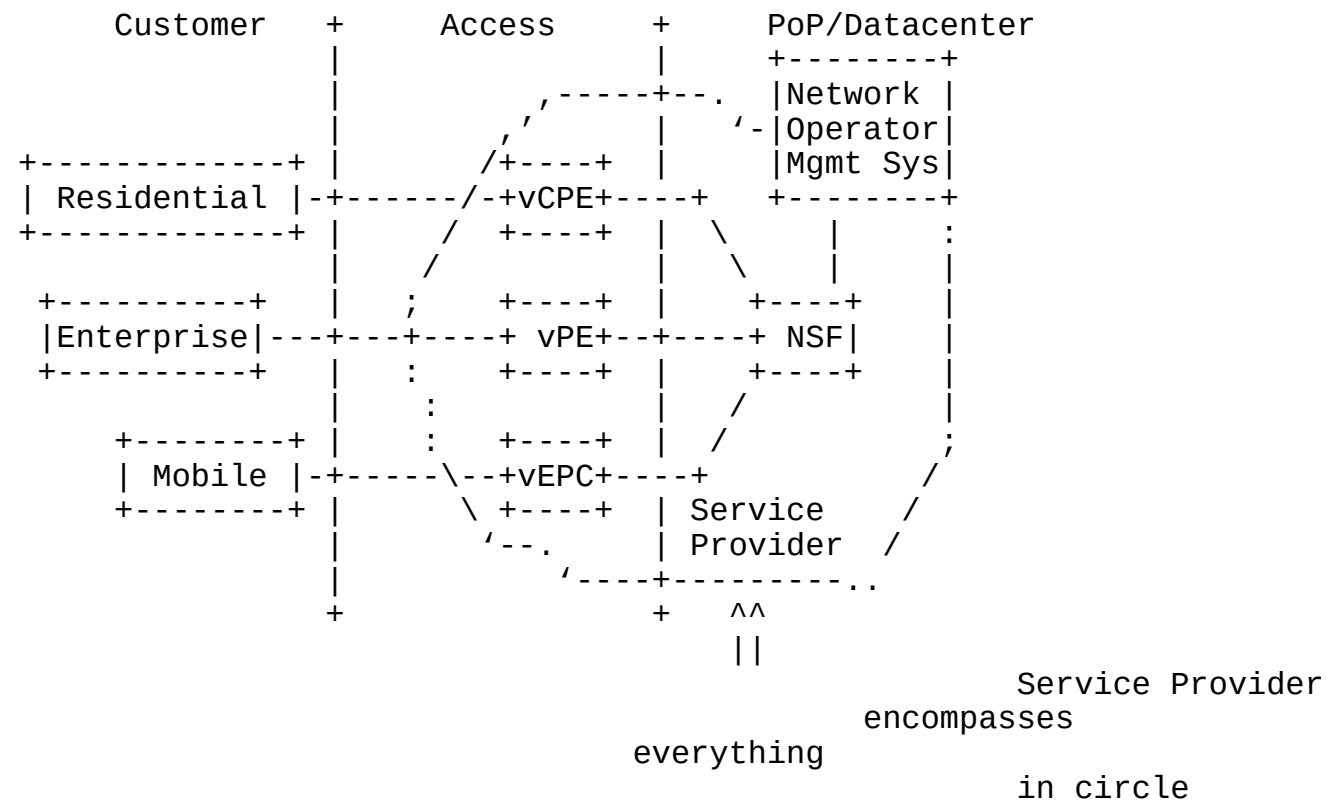

Figure 3: NSF and actors

Different access clients may have different service requests:

Residential: service requests for parental control, content management, and threat management.

Threat content management may include identifying and blocking malicious activities from web contents, mail, or files downloaded. Threat management may include identifying and blocking botnets or malware.

Enterprise: service requests for enterprise flow security policies and managed security services

Flow security policies identify and block malicious activities during access to (or isolation from) web sites or social media applications. Managed security services for an enterprise may

include detection and mitigation of external and internal threats. External threats can include application or phishing attacks, malware, botnet, DDoS, and others.

Service Provider: Service requests for policies that protect service provider networks against various threats (including DDoS, botnets and malware). Such policies are meant to securely and reliably deliver contents (e.g., data, voice, and video) to various customers, including residential, mobile and corporate customers. These security policies are also enforced to guarantee isolation between multiple tenants, regardless of the nature of the corresponding connectivity services.

Mobile: service requests from interfaces which monitor and ensure user quality of experience, content management, parental controls, and external threat management.

Content management for the mobile device includes identifying and blocking malicious activities from web contents, mail, files uploaded/downloaded. Threat management for infrastructure includes detecting and removing malicious programs such as botnet, malware, and other programs that create distributed DoS attacks).

Some access customers may not care about which NSFs are utilized to achieve the services they requested. In this case, provider network orchestration systems can internally select the NSFs (or vNSFs) to enforce the security policies requested by the clients.

Other access customers, especially some enterprise customers, may want to contract separately for dedicated NSFs (most likely vNSFs) for direct control purposes. In this case, here are the steps to associate vNSFs to specific customers:

vNSF Deployment: The deployment process consists in instantiating an 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.

vNSF Customer Provisioning: Once a vNSF is deployed, any customer can subscribe to it. The provisioning life cycle includes the following:

* Customer enrollment and cancellation of the subscription to a vNSF;

* Configuration of the vNSF, based on specific configurations, or derived from common security policies defined by the NSP.

* Retrieval 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.

4.3. Cloud Data Center Scenario

In a data center, network security mechanisms such as firewalls may need to be dynamically added or removed for a number of reasons. These changes may be explicitly requested by the user, or triggered by a pre-agreed upon demand level in the 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 of time frames whenever the bandwidth utilization hits a certain threshold for a specified period. This capacity expansion could result in adding new instances of firewalls on existing machines or provisioning a completely new firewall instance in a different machine.

The on-demand, dynamic nature of security service delivery essentially encourages that the network security "devices" be in software or virtual forms, rather than in a physical appliance form. This requirement 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 in the right domain is of paramount importance, but also in 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, within organization, and inter-organization traffic be separated. Another example is that IPS/IDS services which splits traffic into investment banking traffic and other data traffic to comply with regulatory restrictions for transfer of investment banking information.

4.3.1. 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 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 security policy requirements, which can be numerous. 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 4 shows an example topology of virtual firewalls within a data center.

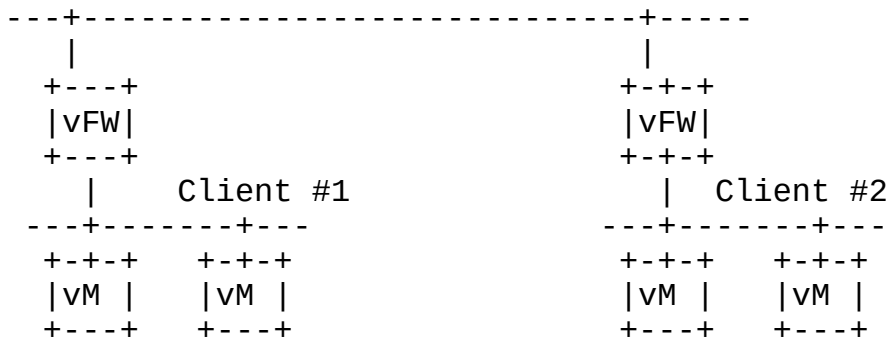

Figure 4: NSF in Data Centers

4.3.2. Firewall Policy Deployment Automation

Firewall rule setting 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 myriads of customers.

Firewall rules today are highly tied with ports and addresses that identify traffic. This makes it very difficult for clients of cloud data centers to construct rules for their own traffic as the clients only see the virtual networks and the virtual addresses. The customer-visible virtual networks and addresses may be different from the actual packets traversing the firewalls (FWs).

Even though most vendors support similar firewall features, the specific rule configuration keywords are different from vendors to vendors, making it difficult for automation. Automation works best when it can leverage a common set of standards that will work across NSFs by multiple vendors and utilize dynamic key management.

4.3.3. Client-Specific Security Policy in Cloud VPNs

Clients of service provider-operated cloud data centers need to secure Virtual Private Networks (VPNs) and virtual security functions that apply the clients' security policies. The security policies may govern communication within the clients' own virtual networks as well as communication 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 (let alone change) what, where and how security policies are implemented on their provider-operated clouds. Indeed, no standards-based framework exists to allow clients to retrieve/ manage security policies in a consistent manner across different providers.

As described above, the dynamic key management is critical for the securing the VPN and the distribution of policies.

4.3.4. Internal Network Monitoring

There are many types of internal traffic monitors that may be managed by a security controller. This includes the class of services referred to as Data Loss Prevention (DLP), or Reputation Protection Services (RPS). Depending on the class of event, alerts may go to internal administrators, or external services.

4.4. Preventing Distributed DoS, Malware and Botnet attacks

In the Internet where everything is connected, preventing unwanted traffic that may cause a Denial of Service (DoS) attack or a distributed DoS (DDoS) attack has become a challenge. Similarly, a network could be exposed to malware attacks and become an attack vector to jeopardize the operation of other networks, by means of remote commands for example. Many networks which carry groups of information (such as Internet of Things (IoT) networks, Information-Centric Networks (ICN), Content Delivery Networks (CDN), Voice over IP packet networks (VoIP), and Voice over LTE (VoLTE)) are also exposed to such remote attacks. There are many examples of remote attacks on these networks, but the following examples will illustrate the issues. A malware attack on an IoT network which carries sensor readings and instructions may attempt to alter the sensor instructions in order to disable a key sensor. A malware attack VoIP or VoLTE networks is software that attempts to place unauthorized long-distance calls. Botnets may overwhelm nodes in ICN and CDN networks so that the networks cannot pass critical data.

In order for organizations to better secure their networks against these kind of attacks, the I2NSF framework should provide a client-side interface that is use case-independent and technology-agnostic. Technology-agnostic is to is defined to be generic, technology independent, and able to support multiple protocols and data models. For example, such an I2NSF interface could be used to provision security policy configuration information that looks for specific malware signatures. Similarly, botnet attacks could be easily prevented by provisioning security policies using the I2NSF client-side interface that prevent access to botnet command and control servers.

4.5. Regulatory and Compliance Security Policies

Organizations must protect their networks against attacks and must also adhere to various industry regulations: any organization that falls under a specific regulation like Payment Card Industry (PCI)-Data Security Standard (DSS) [PCI-DSS] for the payment industry or Health Insurance Portability and Accountability Act [HIPAA] for the healthcare industry must be able to isolate various kinds of traffic. They must also show records of their security policies whenever audited.

The I2NSF client-side interface could be used to provision regulatory and compliance-related security policies. The security controller would keep track of when and where a specific policy is applied and if there is any policy violation; this information can be provided in the event of an audit as a proof that traffic is isolated between specific endpoints, in full compliance with the required regulations.

5. Management Considerations

Management of NSFs usually include the following:

- Life cycle management and resource management of NSFs,
- Device configuration, such as address configuration, device internal attributes configuration, etc.,
- Signaling of events, notifications and changes, and
- Policy rule provisioning.

I2NSF will only focus on the policy provisioning part of NSF management.

6. IANA Considerations

No IANA considerations exist for this document.

7. Security Considerations

Having secure access to control and monitor NSFs is crucial for hosted security services. An I2NSF security controller raises new security threats. It needs to be resilient to attacks and quickly recover from attacks. Therefore, proper secure communication channels have to be carefully specified for carrying controlling and monitoring traffic between the NSFs and their management entity (or entities).

The traffic flow security policies specified by customers can conflict with providers' internal traffic flow security policies. This conflict can be resolved in one of two ways: a) installed policies can restrict traffic if either the customer traffic flow security policies or the provider's internal security policies restrict traffic, or b) can only restrict traffic if both the customer traffic flow security policies and the provider's internal traffic flow security policies restrict data. Either choice could cause potential problems. It is crucial for the management system to flag these conflicts to the customers and to the service provider.

It is important to proper AAA [RFC2904] to authorize access to the network and access to the I2NSF management stream.

Enforcing the appropriate privacy is key to all IETF protocols (see [RFC6973]), and especially important for IETF Security management protocols since they are deployed to protect the network. In some circumstances, security management protocols may be utilized to protect an individual's home, phone, or other personal data. In this case, any solution should carefully consider whether combining management streams abides by the recommendations of [RFC6973] for data minimization, user participation, and security.

8. Contributors

I2NSF is a group effort. The following people actively contributed to the initial use case text: Xiaojun Zhuang (China Mobile), Sumandra Majee (F5), Ed Lopez (Curveball Networks), and Robert Moskowitz (Huawei).

9. Contributing Authors

I2NSF has had a number of contributing authors. The following are contributing authors:

- Linda Dunbar (Huawei),
- Antonio Pastur (Telefonica I+D),
- Mohamed Boucadair (France Telecom),
- Michael Georgiades (Prime Tel),
- Minpeng Qi (China Mobile),
- Shaibal Chakrabarty (US Ignite),
- Nic Leymann (Deutsche Telekom),

- Anil Lohiya (Juniper),
- David Qi (Bloomberg),
- Hyoungshick Kim (Sungkyunkwan University),
- Jung-Soo Park (ETRI),
- Tae-Jin Ahn (Korea Telecom), and
- Se-Hui Lee (Korea Telecom).

10. Acknowledgments

This document was supported by Institute for Information and communications Technology Promotion (IITP) funded by the Korea government (MSIP) [R0166-15-1041, Standard Development of Network Security based SDN].

11. Informative References

[CTA] Cyber Threat Alliance, , "Cyber Threat Alliance", October 2016, <http://cyberthreatalliance.org>.

[EPC-3GPP] Firmin, F., "The Evolved Packet Core", January 2017.

[ETSI-NFV] ETSI GS NFV 002 V1.1.1, , "Network Functions Virtualisation (NFV); Architectural Framework", October 2013.

[Gartner-2013] Messmer, E., "Gartner: Cloud-based security as a service set to take off", October 2013.

[HIPAA] US Congress, , "HEALTH INSURANCE PORTABILITY AND ACCOUNTABILITY ACT OF 1996 (Public Law 104-191)", August 1996, <https://www.hhs.gov/hipaa/>.

[I-D.ietf-i2nsf-gap-analysis] Hares, S., Moskowitz, R., and D. Zhang, "Analysis of Existing work for I2NSF", draft-ietf-i2nsf-gap-analysis-03 (work in progress), March 2017.

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

[I-D.jeong-i2nsf-sdn-security-services]
Jeong, J., Kim, H., Jung-Soo, P., Ahn, T., and s. sehuilee@kt.com, "Software-Defined Networking Based Security Services using Interface to Network Security Functions", draft-jeong-i2nsf-sdn-security-services-05 (work in progress), July 2016.

[I-D.pastor-i2nsf-access-usecases]
Pastor, A. and D. Lopez, "Access Use Cases for an Open OAM Interface to Virtualized Security Services", draft-pastor-i2nsf-access-usecases-00 (work in progress), October 2014.

[I-D.pastor-i2nsf-merged-use-cases]
Pastor, A., Lopez, D., Wang, K., Zhuang, X., Qi, M., Zarny, M., Majee, S., Leymann, N., Dunbar, L., and M. Georgiades, "Use Cases and Requirements for an Interface to Network Security Functions", draft-pastor-i2nsf-merged-use-cases-00 (work in progress), June 2015.

[I-D.qi-i2nsf-access-network-usecase]
Wang, K. and X. Zhuang, "Integrated Security with Access Network Use Case", draft-qi-i2nsf-access-network-usecase-02 (work in progress), March 2015.

[I-D.zarny-i2nsf-data-center-use-cases]
Zarny, M., Leymann, N., and L. Dunbar, "I2NSF Data Center Use Cases", draft-zarny-i2nsf-data-center-use-cases-00 (work in progress), October 2014.

[I-D.zhou-i2nsf-capability-interface-monitoring]
Zhou, C., Xia, L., Boucadair, M., and J. Xiong, "The Capability Interface for Monitoring Network Security Functions (NSF) in I2NSF", draft-zhou-i2nsf-capability-interface-monitoring-00 (work in progress), October 2015.

[NFVUC] ETSI GS NFV 001 V1.1.1, , "ETSI NFV Group Specification, Network Functions Virtualization (NFV) Use Cases", October 2013.

[PCI-DSS] PCI Security Council, , "Payment Card Industry (PCI) Data Security Standard Requirements and Security Assessment Procedures (version 3.2)", April 2016, <https://www.pcisecuritystandards.org/pci_security/>.

[RFC2904] Vollbrecht, J., Calhoun, P., Farrell, S., Gommans, L., Gross, G., de Bruijn, B., de Laat, C., Holdrege, M., and D. Spence, "AAA Authorization Framework", RFC 2904, DOI 10.17487/RFC2904, August 2000, <http://www.rfc-editor.org/info/rfc2904>.

[RFC4948] Andersson, L., Davies, E., and L. Zhang, "Report from the IAB workshop on Unwanted Traffic March 9-10, 2006", RFC 4948, DOI 10.17487/RFC4948, August 2007, <http://www.rfc-editor.org/info/rfc4948>.

[RFC5925] Touch, J., Mankin, A., and R. Bonica, "The TCP Authentication Option", RFC 5925, DOI 10.17487/RFC5925, June 2010, <http://www.rfc-editor.org/info/rfc5925>.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <http://www.rfc-editor.org/info/rfc6973>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <http://www.rfc-editor.org/info/rfc7149>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <http://www.rfc-editor.org/info/rfc7426>.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Phone: +1-734-604-0332
Email: shares@ndzh.com

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

Email: diego.r.lopez@telefonica.com

Myo Zarny
vArmour
800 El Camino Real, Suite 3000
Mountain View, CA 94040
USA

Email: myo@varmour.com

Christian Jacquenet
France Telecom
Rennes, 35000
France

Email: Christian.jacquenet@orange.com

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
USA

Email: rkkumar@juniper.net

Jaehoon Paul Jeong
Department of Software
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon, Gyeonggi-Do 16419
Republic of Korea

Phone: +82 31 299 4957
Fax: +82 31 290 7996
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php