I2NSF BOF
Internet-Draft
Intended status: Standards Track
Expires: April 20, 2016

S. Hares
Huawei
A. Pastor
Telefonica I+D
K. Wang
China Mobile
D. Zhang

M. Zarny
Goldman Sachs
October 18, 2015

Analysis of Use Cases and Gaps in Technology for I2NSF
draft-hares-i2nsf-use-case-gap-analysis-00.txt

Abstract

This document provides a summary of the I2NSF use cases plus a summary of the stat of the art in industries and IETF work which is relevant to the Interface to Network Security Function (I2NSF). The I2NSF focus is to define data models and interfaces in order to control and monitor the physical and virtual aspects of network security functions. The use cases are 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."

This Internet-Draft will expire on April 20, 2016.

Table of Contents

1. Introduction

Enterprise, residential, and mobile customers 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 provides an analysis of the use cases, gaps analysis of existing technology, recommendations for requirements for I2NSF, and security considerations.

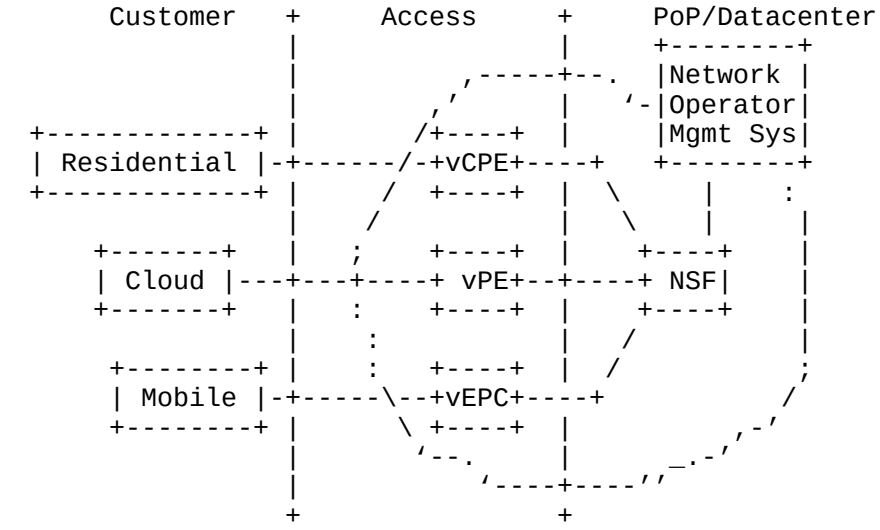


Figure 1: NSF and actors

1.1. What is I2NSF

A Network Security Function (NSF) is a function used to ensure integrity, confidentiality, or availability of network communications, to detect unwanted network activity, or to block or at least mitigate the effects of unwanted activity. NSFs are provided and consumed in increasingly diverse environments. Users could consume network security services enforced by NSFs hosted by one or more providers - which may be their own enterprise, service providers, or a combination of both. Similarly, service providers may offer their customers network security services that are enforced by multiple security products, functions from different vendors, or open source technologies. NSFs may be provided by physical and/or virtualized infrastructure. Without standard interfaces to control and monitor the behavior of NSFs, it has become virtually impossible for providers of security services to automate service offerings that utilize different security functions from multiple vendors.

1.2. I2NSF Standarization

The Interface to NSF devices (I2NSF) work proposes to standardize a set of software interfaces and data modules to control and monitor the physical and virtual NSFs. Since different security vendors support different features and functions, the I2NSF will focus on the flow-based NSFs that provide treatment to packets or flows such found

in IPS/IDS devices, web filtering devices, flow filtering devices, deep packet inspection devices, pattern matching inspection devices, and re-mediation devices.

There are two layers of interfaces envisioned in the I2NSF approach:

- The I2NSF Capability Layer specifies how to control and monitor NSFs at a functional implementation level. This the focus for this phase of the I2NSF Work.

- The I2NSF Service Layer defines how the security policies of clients may be expressed and monitored.

For the I2NSF capability layer, the I2NSF work proposes an interoperable protocol that passes NSF provisioning rules and orchestration information between I2NSF client on a network manager and I2NSF agent on an NSF device. It is envisioned that clients of the I2NSF interfaces include management applications, service orchestration systems, network controllers, or user applications that may solicit network security resources.

The I2NSF work to define this protocol includes the following work:

- defining an informational model that defines the concepts for standardizing the control and monitoring of NSFs,

- defining a set of Yang data models from the information model that identifies the data that must be passed,

- creating a capability registry (an IANA registry) that identifies the characteristics and behaviours of NSFs in vendor-neutral vocabulary without requiring the NSFs to be standardized.

- examining existing secure communication mechanisms to identify the appropriate ones for carrying the data that provisions and monitors information between the NSFs and their management entity (or entities).

1.3. Structure of the document

This document reviews the terminology (section 3), analyzes the use cases (section 4) and gaps in current technology (section 5), recommends certain requirements for I2NSF protocol(section 6), and discusses security consideration (section 8).

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.

- type of NSFs: NSFs considered in this draft include virtualized and non-virtualized NSFs.

- Cloud DC: A data center that is not on premises of enterprises, but has compute/storage resources that can be requested or purchased by the enterprises. The enterprise is actually getting a virtual data center. The Cloud Security Alliance (CSA) (http://cloudsecurityalliance.org) focus on adding security to this environment. A specific research topic is security as a service within the cloud data center.

- Cloud-based security functions: Network Security Function (NSF) hosted and managed by service providers or different administrative entity.

- DC: Data Center

- Domain: The term Domain in this draft has the following different connotations in different scenarios:

  * Client--Provider relationship, i.e. client requesting some network security functions from its provider;

  * Domain A - Domain B relationship, i.e. one operator domain requesting some network security functions from another operator domain; or

  * Applications -- Network relationship, i.e. an application (e.g. cluster of servers) requesting some functions from network, etc.

  The domain context is important because it indicates the interactions the security is focused on.

- o I2NSF agent - a piece of software in a device that implements a network security function which receives provisioning information and requests for operational data (monitoring data) across the I2NSF protocol from an I2NSF client.

- o I2NSF client - A security client software that utilizes the I2NSF protocol to read, write or change the provisioning network security device via software interface using the I2NSF protocol (denoted as I2RS Agent)

- o I2NSF Management System - I2NSF client operates within an network management system which serves as a collections and distribution point for security provisioning and filter data. This management system is denoted as I2NS management system in this document.

- o Virtual Security Function: a security function that can be requested by one domain but may be owned or managed by another domain.

4. Use Cases

This section discusses general use cases, access use cases, and cloud use cases.

4.1. 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:

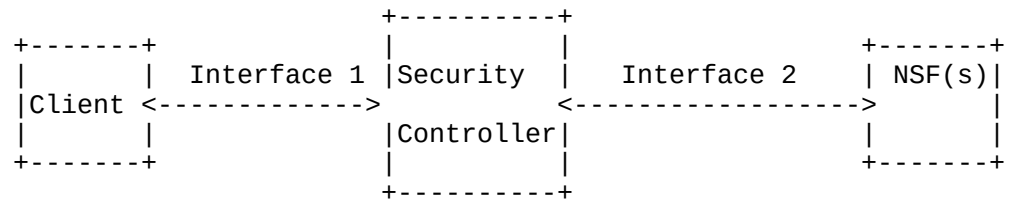


Figure 2: Interaction between Entities

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.

4.1.1. 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 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.

4.1.2. 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.

4.1.3. 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.

4.1.4. 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

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

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.

4.2. Access Networks

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).

4.2.1. 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.

4.2.2. vNSF Customer Provisioning

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

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.

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.

4.3. 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.

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 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.

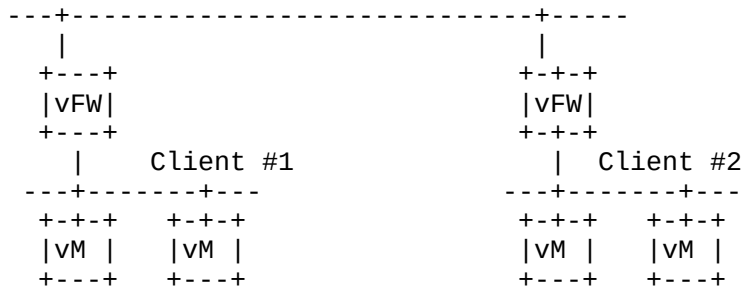


Figure 3: NSF in DataCenter

4.3.2. 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.

4.3.2.1. 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.

4.4. 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:

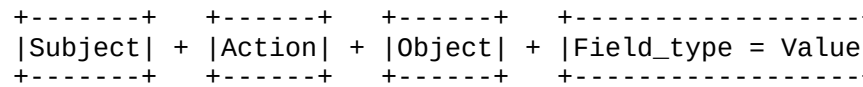


Figure 4: High-Level Security Policy Format

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"

4.4.1. 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:

- Identity Management: Includes all services related with identity, authentication and key management. Some examples are:
  - AAA (Authentication, Authorization, Accounting) services
  - Remote identity management
  - Remote identity management
- 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

5. Gap Analysis

5.1. Structure of the gap analysis

This document provides a analysis of the gaps in the state of art in the following industry forums:

IETF working groups (section 5.2)

ETSI Network Functions Virtualization Industry Specification Group (ETSI NFV ISG), (section 5.3)

OPNFV Open Source Group (section 5.4)

Open Stack - Firewall as a service (OpenStack Firewall FaaS) (section 5.5) (http://docs.openstack.org/admin-guide-cloud/content/install_neutron-fwaas-agent.html)

Cloud Security Alliance Security (CSA)as a Service (section 5.6) (https://cloudsecurityalliance.org/research/secaas/#_overview)

In-Depth Review of Some IETF Protocols (section 5.7)

5.2. IETF Gap analysis

The IETF gap analysis first examines the IETF mechanisms which have been developed to secure the IP traffic flows through a network. Traffic filters have been defined by IETF specifications at the access points, the middle-boxes, or the routing systems. Protocols have been defined to carry provisioning and filtering traffic between a management system and an IP system (router or host system). Current security work (SACM working group (WG), MILE WG, and DOTS WG) is providing correlation of events monitored with the policy set by filters. This section provides a review the filter work, protocols, and security correlation for monitors.

5.2.1. Traffic Filters

5.2.1.1. Overview

The earliest filters defined by IETF were access filters which controlled the acceptance of IP packet data flows. Additional policy filters were created as part of the following protocols:

- COPS protocol [RFC2748] for controlling access to networks,
- Next steps in Signalling (NSIS) work (architecture: [RFC4080] protocol: [RFC5973]), and
- the Port Control Protocol (PCP) to enables IPv4 to IPv6 flexible address and port mapping for NATs and Firewalls,

Today NETMOD and I2RS Working groups are specifying additional filters in Yang modules to be used as part of the NETCONF or I2RS enhancement of NETCONF/RESTCONF.

The routing filtering is outside the scope of the flow filtering, but flow filtering may be impacted by route filtering. An initial model for the routing policy is in [I-D.shaikh-rtgwg-policy-model]

This section provides an overview of the flow filtering as an introduction to the I2NSF GAP analysis. Additional detail on NETCONF, NETMOD, I2RS, PCP, and NSIS is available in the Detailed I2NSF analysis.

5.2.1.1.1. Data Flow Filters in NETMOD and I2RS

The current work on expanding these filters is focused oncombining a configuration and monitoring protocol with Yang data models. [I-D.ietf-netmod-acl-model] provides a set of access lists filters which can permit or deny traffic flow based on headers at the MAC, IP layer, and Transport layer. The configuration and monitoring protocols which can pass the filters are: NETCONF protocol [RFC6241], RESTCONF [I-D.ietf-netconf-restconf], and the I2RS protocol. The NETCONF and RESTCONF protocols install these filters into forwarding tables. The I2RS protocol uses the ACLs as part of the filters installed in an ephemeral protocol-independent filter-based RIB [I-D.kini-i2rs-fb-rib-info-model] which controls the flow of traffic on interfaces specifically controlled by the I2RS filter-based FIB.

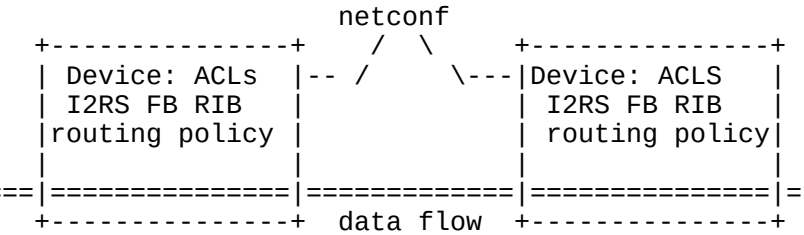


Figure 5 -I2RS Filter-Based RIB

The I2RS protocol is a programmatic interface to the routing system. At this time, the I2RS is targeted to be extensions to the NETCONF/ RESTCONF protocols to allow the NETCONF/RESTCONF protocol to support a highly programmatic interface with high bandwidth of data, highly reliable notifications, and ephemeral state (see [I-D.ietf-i2rs-architecture]). Please see the background section on I2RS for additional details on the requirements for this extension to the NETCONF/RESTCONF protocol suite.

The vocabulary set in [I-D.ietf-netmod-acl-model] is limited, so additional protocol independent filters were written for the I2RS Filter-Based RIBs in [I-D.hares-i2rs-bnp-eca-data-model], and protocol specific filters for SFC [I-D.dunbar-i2rs-discover-traffic-rules].

One thing important to note is that NETCONF and RESTCONF manage device layer yang models. However, as figure 6 shows, there are multiple device level, network-wide level, and application level yang modules. The access lists defined by the device level forwarding table may be impacted by the routing protocols, the I2RS ephemeral protocol independent Filter-Based FIB, or some network-wide security issue (IPS/IDS).

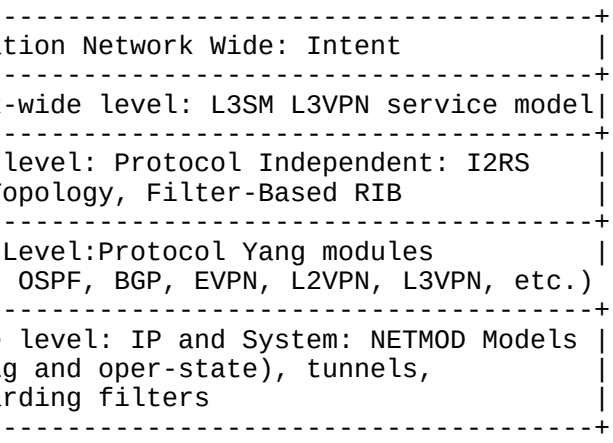


Figure 6 levels of Yang modules

5.2.1.1.2. I2NSF Gap analysis

The gap is that none of the current work on these filters considers all the variations of data necessary to do IPS/IDS, web-filters, stateful flow-based filtering, security-based deep packet inspection, or pattern matching with re-mediation. The I2RS Filter-Based RIB work is the closest associated work, but the focus has not been on IDS/IPS, web-filters, security-based deep packet inspection, or pattern matching with re-mediation.

The I2RS Working group (I2RS WG) is focused on the routing system so security expertise for these IDP/IPS, Web-filter, security-based deep-packet inspection has not been targeted for this WG.

Another gap is there is no capability registry (an IANA registry) that identifies the characteristics and behaviours of NSFs in vendor-neutral vocabulary without requiring the NSFs to be standardized.

What I2NSF can use from NETCONF/RESTCONF and I2RS

I2NSF should consider using NETCONF/RESTCONF protocol and the I2RS proposed enhancement to the NETCONF/RESTCONF protocol.

5.2.1.2. Middle-box Filters

5.2.1.2.1. Midcom

Midcom Summary: MIDCOM developed the protocols for applications to communicate with middle boxes. However, MIDCOM have not used by the industry for a long time. This is because there was a lot of IPR

encumbered technology and IPR was likely a bigger problem for IETF than it is today. MIDCOM is not specific to SIP. It was very much oriented to NAT/FW devices. SIP was just one application that needed the functionality. MIDCOM is reservation-oriented and there was an expectation that the primary deployment environment would be VoIP and real-time conferencing, including SIP, H.323, and other reservation-oriented protocols. There was an assumption that there would be some authoritative service that would have a view into endpoint sessions and be able to authorize (or not) resource allocation requests. In other word, there's a trust model there that may not be applicable to endpoint-driven requests without some sort of trusted authorization mechanisms/tools. Therefore, there is a specific information model applied to security devices, and security device requests, that was developed in the context of an SNMP MIB. There is also a two-stage reservation model, which was specified in order to allow better resource management.

Why I2NSF is different than Midcom

MIDCOM is different than I2NSF because its SNMP scheme doesn't work with the virtual network security functions (vNSF) management.

MidCom RFCs:

[RFC3303] - Midcom architecture

[RFC5189] - Midcom Protocol Semantics

[RFC3304] - Midcom protocol requirements

5.2.1.3. Security Work

5.2.1.3.1. Overview

Today's NSFs in security devices can handle flow-based security by providing treatment to packets/flows, such as IPS/IDS, Web filtering, flow filtering, deep packet inspection, or pattern matching and re-mediation. These flow-based security devices are managed and provisioned by network management systems.

No standardized set of interoperable interfaces control and manage the NSFs so that a central management system can be used across security devices from multiple Vendors. I2NSF work plan is to standardize a set of interfaces by which control and management of NSFs may be invoked, operated, and monitored by:

creating an information model that defines concepts required for standardizing the control and monitoring of NSFs, and from the

information model create data models. (The information model will be used to get early agreement on key technical points.)

creating a capability registry (at IANA) that enables the characteristics and behavior of NSFs to be specified using a vendor-neutral vocabulary without requiring the NSFs themselves to be standardized.

define the requirements for an I2NSF protocol to pass this traffic. (Hopefully re-using existing protocols.)

The flow-filtering configuration and management must fit into the existing security area's work plan. This section considers how the I2NSF fits into the security area work under way in the SACM (security automation and control), DOTS (DDoS Open Threat Signalling), and MILE (Management Incident Lightweight Exchange).

5.2.1.3.2. Security Work and Filters

In the proposed I2NSF work plan, the I2NSF security network management system controls many NSF nodes via the I2NSF Agent. This control of data flows is similar to the COPS example in section 5.7.4.

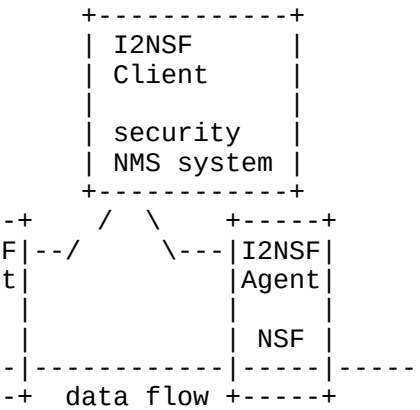


Figure 7

The other security protocols work to interact within the network to provide additional information in the following way:

o SACM [I-D.ietf-sacm-architecture] describes an architecture which tries to determine if the end-point security policies and the reality (denoted as security posture) align.

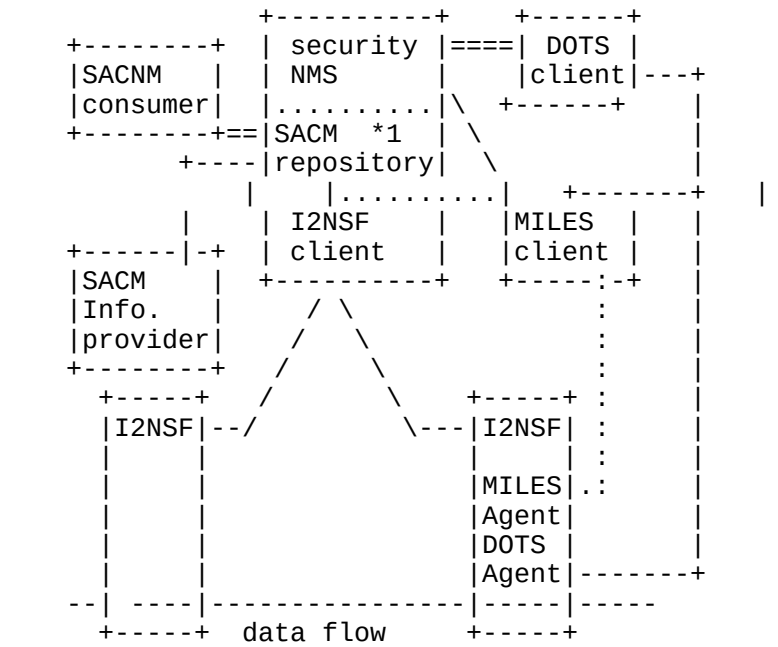
[I-D.ietf-sacm-terminology] defines posture as the configuration and/or status of hardware or software on an endpoint as it pertains to an organization's security policy. Filters can be considered on the configuration or status pieces that needs to be monitored.

- DOTS (DDoS Open Threat Signalling) - is working on coordinating the mitigation of DDoS attacks. A part of DDoS attach mitigation is to provide lists of addresses to be filtered via IP header filters.

- MILE (Managed Incident LIghtweight Exchange) - is working on creating a standardized format for incident and indicator reports, and creating a protocol to transport this information. The incident information MILE collects may cause changes in data-flow filters on one or more NSFs.

5.2.1.3.3. I2NSF interaction

The network management system that the I2NSF client resides on may interact with other clients or agents developed for the work ongoing in the SACM, DOTS, and MILES working groups. This section describes how the addition of I2NSF's ability to control and monitor NSF devices is compatible and synergistic with these existing efforts.



*1 - this is the SACM Controller (CR) with its broker/proxy/repository show as described in the SACM architecture.

Figure 8

Figure 8 provides a diagram of a system the I2NSF, SACM, DOTS and MILES client-agent or consumer-broker-provider are deployed together. The following are possible positive interactions these scenario might have:

- An security network management system (NMS) can contain a SACM repository and be connected to SACM information provider and a SACM consumer. The I2NSF may provide one of the ways to change the forwarding filters.

- The security NMS may also be connected to DOTS DDoS clients managing the information and configuring the rules. The I2NSF may provide one of the ways to change forwarding filters.

- The MILES client on a security network management system talking to the MILES agent on the node may react to the incidents by using I2NSF to set filters. DOTS creates black-lists, but does not have a complete set of filters.

5.2.1.3.4. Benefits from the Interaction

I2NSF's ability to provide a common interoperable and vendor neutral interface may allow the security NMS to use a single change to change filters. SACM provides an information model to describe end-points, but does not link this directly to filters.

DOTS creates black-lists based on source and destination IP address, transport port number, protocol ID, and traffic rate. Like NETMOD's, ACLS are not sufficient for all filters or control desired by the NSF boxes.

The incident data captured by MILES will not have enough filter information to provide NSF devices with general services. The I2NSF will be able to handle the MILE incident data and create alerts or reports for other security systems.

5.3. ETSI NFV

5.3.1. ETSI Overview

Network Function Virtualization (NFV) provides the service providers with flexibility, cost effective and agility to offer their services to customers. One such service is the network security function which guards the exterior of a service provider or its customers.

The flexibility and agility of NFV encourages service providers to provide different products to address business trends in their market to provide better service offerings to their end user. A traditional product such as the network security function (NSF) may be broken into multiple virtual devices each hosted from another vendor. In the past, network security devices may have been single sourced from a small set of vendors - but in the NFV version of NSF devices, this reduced set of sources will not provide a competitive edge. Due to this market shift, the network security device vendors are realizing that the proprietary provisioning protocols and formats of data may be a liability. Out of the NFV work has arisen a desire for a single interoperable network security device provisioning and control protocol.

The I2NSF will be deployed along networks using other security and NFV technology. As section 3 described, the NFV NSF security is deployed along side other security functions (AAA, SACM, DOTS, and MILE devices) or deep-packet-inspection. The ETSI Network Functions Virtualization: NFV security: Security and Trust guidance document (ETSI NFV SEC 003 1.1.1 (2014-12)) indicates that multiple administrative domains will deployed in carrier networks. One example of these multiple domains is hosting of multiple tenant

domains (telecom service providers) on a single infrastructure domain (infrastructure service) as figure 9 shows. The ETSI Inter inter-VNFM document (aka Ve-Vnfn) between the element management system and the Virtual network function is the equivalent of the interface between the I2NSF client on a management system and the I2NSF agent on the network security feature VNF.

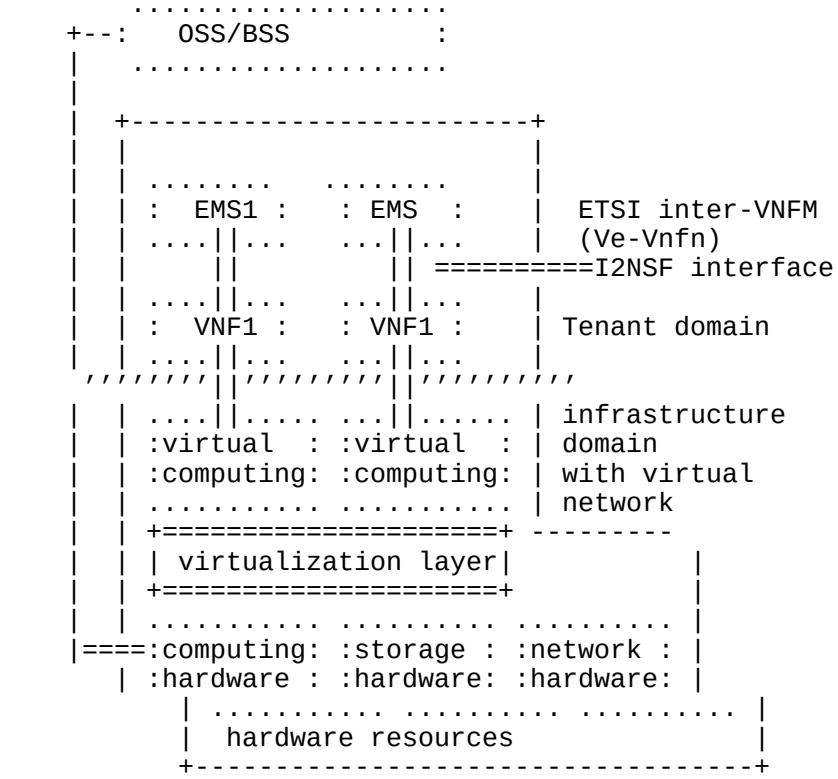


Figure 9

The ETSI proof of concept work has worked on the following security proof of concepts:

o #16 - NFVIaas with Secure, SDN controlled WAN Gateway,

5.3.2. I2NSF Gap Analysis

The I2NSF will be deployed on top of virtual computing linked together by virtual routers configured by NETCONF/RESTCONF or I2RS which provision and monitoring the L1, L2, l3 and service pathways through the network.

In the NFV-related productions, the current architecture does not have a protocol to maintain an interoperability provisioning from I2NSF client to I2NSF agent. The result is that service providers have to manage the interoperability using private protocols. In response to this problem, the device manufacturers and the service providers have begun to discuss an I2NSF protocol for interoperable passing of provisioning and filter in formation.

Open source work (such as OPNFV) provides a common code base for providers to start their NFV work from. However, this code base faces the same problem. There is no defacto standard protocol.

5.4. OPNFV

The OPNFV (www.opnfv.org) is a carrier-grade integrated, open source platform focused on accelerating the introduction of new Network Function Virtualization (NFV) products and service. The OPNFV Moon project is focused on adding the security interface for a network management system within the Tenant NFVs and the infrastructure NFVs (as shown in figure 4). This section provides an overview of the OPNFV Moon project and a gap analysis between I2NSF and the OPNFV Moon Project.

5.4.1. OPNFV Moon Project

The OPNFV moon project (https://wiki.opnfv.org) is a security management system. NFV uses cloud computing technologies to virtualize the resources and automate the control. The Moon project is working on a security manager for the Cloud computing infrastructure (https://wiki.opnfv.org/moon). The Moon project proposes to provision a set of different cloud resources/services for VNFs (Virtualized Network Functions) while managing the isolation of VNS, protection of VNFs, and monitoring of VNS. Moon is creating a security management system for OPNFV with security managers to protect different layers of the NFV infrastructure. The Moon project is choosing various security project mechanisms "a la cart" to enforcement related security managers. A security management system integrates mechanisms of different security aspects. This project will first propose a security manager that specifies users’ security requirements. It will also enforce the security managers through various mechanisms like authorization for access control, firewall for networking, isolation for storage, logging for tractability, etc.

The Moon security manager operates a VNF security manager at the ETSI VeVnfm level where the I2NSF protocol is targeted as figure 10 shows. Figure 10 also shows how the OPNFV VNF Security project mixes the I2NSF level with the device level.

The Moon project lists the following gaps in OpenStack:

- No centralized control for compute, storage, and networking. Open Stack uses Nova for computing and Swift for software. Each system has a configuration file and its own security policy. This lacks the synchronization mechanism to build a complete secure configuration for OPNF.

- No dynamic control so that if a user obtains the token, the is no way to obtain control over the user.

- No customization or flexibility to allow integration into different vendors,

- No fine grain authorization at user level. Authorization is only at the API

Moon addresses these issues adding authorization, logging, IDS, enforcement of network policy, and storage protection. Moon is based on OpenStack Keystone.

Deliverable time frame: 2S 2015

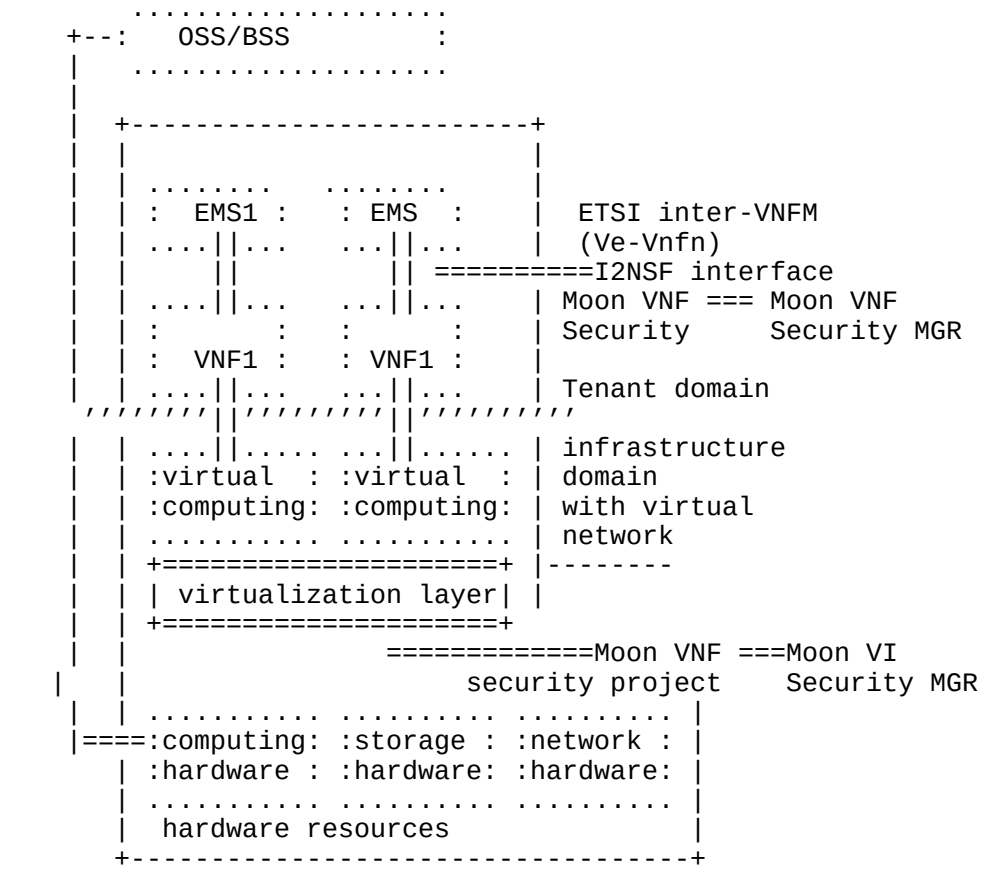


Figure 10

5.4.2. Gap Analysis for OPNFV Moon Project

OpenStack congress does not provide vendor independent systems.

5.5. OpenStack Security Firewall

OpenStack has advanced features of: a) API for managing security groups (http://docs.openstack.org/admin-guide-cloud/content/section_securitygroups.html) and b) firewalls as a service (http://docs.openstack.org/admin-guide-cloud/content/fwaas_api_abstractions.html).

This section provides an overview of this open stack work, and a gap analysis of how I2NSF provides additional functions

5.5.1. Overview of API for Security Group

The security group with the security group rules provides ingress and egress traffic filters based on port. The default group drops all ingress traffic and allows all egress traffic. The groups with additional filters are added to change this behaviour. To utilize the security groups, the networking plug-in for Open Stack must implement the security group API. The following plug-ins in OpenSTsack currently implement this security: ML2, Open vSwitch, Linux Bridge, NEC, and VMware NSX. In addition, the correct firewall driver must be added to make this functional.

5.5.2. Overview of Firewalls as a Service

Firewall as a service is an early release of an API that allows early adopters to test network implementations. It contains APIs with parameters for firewall rules, firewall policies, and firewall identifiers. The firewall rules include the following information:

- identification of rule (id, name, description)
- identification tenant rule associated with,
- links to installed firewall policy,
- IP protocol (tcp, udp, icmp, none)
- source and destination IP address
- source and destination port
- action: allow or deny traffic
- status: position and enable/disabled

The firewall policies include the following information:

- identification of the policy (id, name, description),
- identification of tenant associated with,
- ordered list of firewall rules,
- indication if policy can be seen by tenants other than owner, and
- indication if firewall rules have been audited.

The firewall table provides the following information:

- identification of firewall (id, name, description),
- tenant associated with this firewall,
- administrative state (up/down),
- status (active, down, pending create, pending delete, pending update, pending error)
- firewall policy ID this firewall is associated with

5.5.3. I2NSF Gap analysis

The OpenStack work is preliminary (security groups and firewall as a service). This work does not allow any of the existing network security vendors provide a management interface. Security devices take time to be tested for functionality and their detection of security issues. The OpenStack work provides an interesting simple set of filters, and may in the future provide some virtual filter service. However, at this time this open source work does not address the single management interfaces for a variety of security devices.

I2NSF is proposing rules that will include Event-Condition-matches (ECA) with the following matches

packet based matches on L2, L3, and L4 headers and/or specific addresses within these headers,

context based matches on schedule state and schedule, [Editor: Need more details here.]

The I2NSF is proposing action for these ECA policies of:

basic actions of deny, permit, and mirror,

advanced actions of: IPS signature filtering and URL filtering.

5.6. CSA Secure Cloud

5.6.1. CSA Overview

The Cloud Security Alliance (CSA)(www.cloudsecurityaliance.org) defined security as a service (SaaS) in their Security as a Service working group (SaaS WG) during 2010-2012. The CSA SaaS group defined ten categories of network security (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_V1_0.pdf) and provides implementation guidance for each of

these ten categories This section provides an overview of the CSA SaaS working groups documentation and a Gap analysis for I2NSF

5.6.1.1. CSA Security as a Service(SaaS)

The CSA SaaS working group defined the following ten categories, and provided implementation guidance on these categories:

1. Identity Access Management (IAM) (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_1_IAM_Implementation_Guidance.pdf)

2. Data Loss Prevention (DLP) (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_2_DLP_Implementation_Guidance.pdf)

3. Web Security (web) (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_3_Web_Security_Implementation_Guidance.pdf),

4. Email Security (email) (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_4_Email_Security_Implementation_Guidance.pdf),

5. Security Assessments (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_5_Security_Assessments_Implementation_Guidance.pdf),

6. Intrusion Management (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_6_Intrusion_Management_Implementation_Guidance.pdf),

7. Security information and Event Management (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_7_SIEM_Implementation_Guidance.pdf),

8. Encryption (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_8_Encryption_Implementation_Guidance.pdf),

9. Business Continuity and Disaster Recovery (BCDR) https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_9_BCDR_Implementation_Guidance.pdf), and

10. Network Security (https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_10_Network_Security_Implementation_Guidance.pdf).

The sections below give an overview these implementation guidances

5.6.1.2. Identity Access Management (IAM)

document:
(https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_1_IAM_Implementation_Guidance.pdf)

The identity management systems include the following services:

- Centralized Directory Services,
- Access Management Services,
- Identity Management Services,
- Identity Federation Services,
- Role-Based Access Control Services,
- User Access Certification Services,
- Privileged User and Access Management,
- Separation of Duties Services, and
- Identity and Access Reporting Services.

The IAM device communications with the security management system that controls the filtering of data. The CSA SaaS IAM specification states that interoperability between IAM devices and secure access network management systems is a a problem. This 2012 implementation report confirms there is a gap with I2NSF

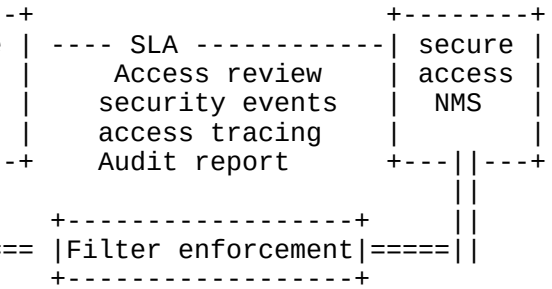


Figure 11

5.6.1.3. Data Loss Prevention (DLP)

Document:
(https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_2_DLP_Implementation_Guidance.pdf)

The data loss prevention (DLP)services must address:

- origination verification,
- integrity of data,
- confidentiality and access control,
- accountability,
- avoiding false positives on detection, and
- privacy concerns.

The CSA SaaS DLP device communications require that it have the enforcement capabilities to do the following:

alert and log data loss,

delete data on system or passing through,

filter out (block/quarantine) data,

reroute data,

encrypt data

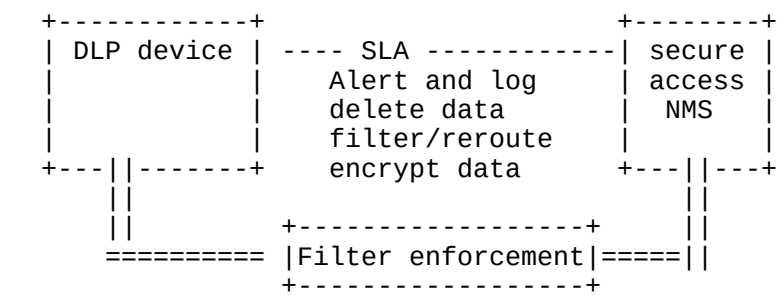


Figure 12

5.6.1.4. Web security(Web))

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_3_Web_Security_Implementation_Guidance.pdf

The web security services must address:

- Web 2.0/Social Media controls,
- Malware and Anti-Virus controls,
- Data Loss Prevention controls (over Web-based services like Gmail or Box.net),
- XSS, JavaScript and other web specific attack controls
- Web URL Filtering,
- Policy control and administrative management,
- Bandwidth management and quality of service (QoS) capability, and
- Monitoring of SSL enabled traffic.

The CSA SaaS Web services device communications require that it have the enforcement capabilities to do the following:

alert and log malware or anti-virus data patterns,

delete data (malware and virus) passing through systems,

filter out (block/quarantine) data,

filter Web URLs,

interact with policy and network management systems,

control bandwidth and QoS of traffic, and

monitor encrypted (SSL enabled) traffic,

All of these features either require the I2NSF standardized I2NSF client to I2NSF agent to provide multi-vendor interoperability.

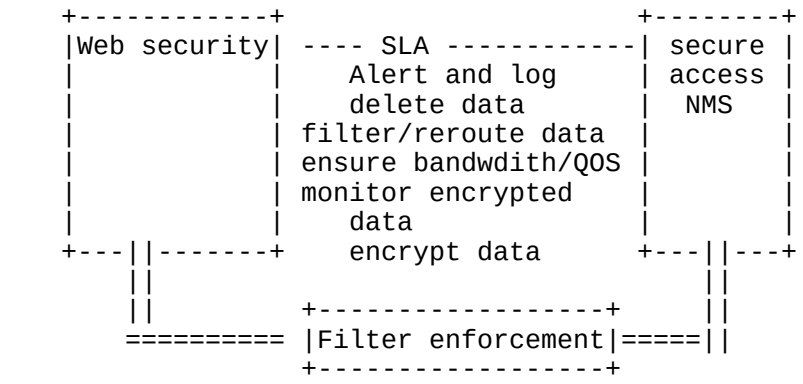


Figure 13

5.6.1.5. Email Security (email))

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_4_Email_Security_Implementation_Guidance.pdf

The CSA Document recommends that email security services must address:

- Common electronic mail components,
- Electronic mail architecture protection,
- Common electronic mail threats,
- Peer authentication,
- Electronic mail message standards,
- Electronic mail encryption and digital signature,
- Electronic mail content inspection and filtering,
- Securing mail clients, and
- Electronic mail data protection and availability assurance techniques

The CSA SaaS Email security services requires that it have the enforcement capabilities to do the following:

provide the malware and spam detection and removal,

alert and provide rapid response to email threats,

identify email users and secure remote access to email,

do on-demand provisioning of email services,

filter out (block/quarantine) email data,

know where the email traffic or data is residing (to to regulatory issues), and

be able to monitor encrypted email,

be able to encrypt email,

be able to retain email records (while abiding with privacy concerns), and

interact with policy and network management systems.

All of these features require the I2NSF standardized I2NSF client to I2NSF agent to provide multi-vendor interoperability.

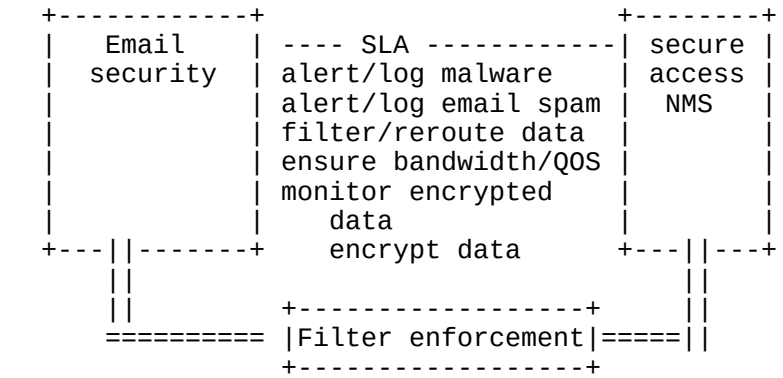


Figure 14

5.6.1.6. Security Assessment

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_5_Security_Assessments_Implementation_Guidance.pdf

The CSA SaaS Security assessment indicates that assessments need to be done on the following devices:

o hypervisor infrastructure,

- o network security compliance systems,
- o Servers and workstations,
- o applications,
- o network vulnerabilities systems,
- o internal auditor and intrusion detection/prevention systems (IDS/IPS), and
- o web application systems.

All of these features require the I2NSF working group standardize the way to pass these assessments to and from the I2NSF client on the I2NSF management system and the I2NSF Agent.

5.6.1.7. Intrusion Detection

Document: https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_6_Intrusion_Management_Implementation_Guidance.pdf)

The CSA SaaS Intrusion detection management includes intrusion detection through: devices:

- o Network traffic inspection, behavioural analysis, and flow analysis,
- o Operating System, Virtualization Layer, and Host Process Events monitoring,
- o monitoring of Application Layer Events, and
- o Correlation Techniques, and other Distributed and Cloud-Based Capabilities

Intrusion response includes both:

- o Automatic, Manual, or Hybrid Mechanisms,
- o Technical, Operational, and Process Mechanisms.

The CSA SaaS recommends the intrusion security management systems include provisioning and monitoring of all of these types of intrusion detection (IDS) or intrusion protection devices. The management of these systems requires also requires:

Central reporting of events and alerts,

administrator notification of intrusions,

Mapping of alerts to Cloud-Layer Tenancy,

Cloud sourcing information to prevent false positives in detection, and

allowing for redirection of traffic to allow remote storage or transmission to prevent local evasion.

All of these features require the I2NSF standardized I2NSF client to I2NSF agent to provide multi-vendor interoperability.

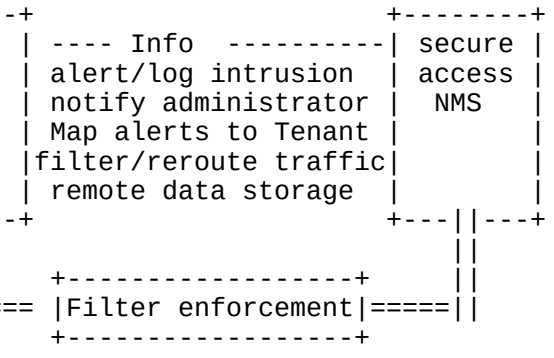


Figure 15

5.6.1.8. Security Information and Event Management(SEIM)

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_7_SIEM_Implementation_Guidance.pdf)

The Security Information and Event Management (SEIM) receives data from a wide range of security systems such as Identity management systems (IAM), data loss prevention (DLP), web security (Web), email security (email), intrusion detection/prevision (IDS/IPS)), encryption, disaster recovery, and network security. The SEIM combines this data into a single streams. All the requirements for data to/from these systems are replicated in these systems needs to give a report to the SIEM system.

A SIEM system would be prime candidate to have a I2NSF client that gathers data from an I2NSF Agent associated with these various types of security systems. The CSA SaaS SIEM functionality document

suggests that one concern is to have standards that allow timely recording and sharing of data. I2NSF can provide this.

5.6.1.9. Encryption

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_8_Encryption_Implementation_Guidance.pdf

The CSA SaaS Encryption implementation guidance document considers how one implements and manages the following security systems:

key management systems (KMS), control of keys, and key life cycle;

Shared Secret encryption (Symmetric ciphers),

No-Secret or Public Key Encryption (asymmetric ciphers),

hashing algorithms,

Digital Signature Algorithms,

Key Establishment Schemes,

Protection of Cryptographic Key Material (FIPS 140-2; 140-3),

Interoperability of Encryption Systems, Key Conferencing, Key Escrow Systems, and others

application of Encryption for Data at rest, data in transit, and data in use;

PKI (including certificate revocation "CRL");

Future application of such technologies as Homomorphic encryption, Quantum Cryptography, Identitybased Encryption, and others;

Crypto-system Integrity (How bad implementations can under mind a crypto-system), and

Cryptographic Security Standards and Guidelines

The wide variety of encryption services require the security management systems be able to provision, monitor, and control the systems that are being used to encrypt data. This document indicates in the implementation sections that the standardization of interfaces to/from management systems are key to good key management systems, encryption systems, and crypto-systems.

5.6.1.10. Business Continuity and Disaster Recovery (BC/DR)

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_9_BCDR_Implementation_Guidance.pdf

The CSA SaaS Business Continuity and Disaster Recovery (BC/DR) implementation guidance document considers the systems that implement the the contingency plans and measures designed and implemented to ensure operational resiliency in the event of any service interruptions. BC/DR systems includes:

Business Continuity and Disaster Recovery BC/DR as a service, including categories such as complete Disaster Recovery as a Service (DRaaS), and subsets such as file recovery, backup and archive,

Storage as a Service including object, volume, or block storage;

old Site, Warm Site, Hot Site backup plans;

IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service);

Insurance (and insurance reporting programs)

Business Partner Agents (business associate agreements);

System Replication (for high availability);

Fail-back to Live Systems mechanisms and management;

Recovery Time Objective (RTO) and Recovery Point Objective (RPO);

Encryption (data at rest [DAR], data in motion [DIM], field level);

Realm-based Access Control;

Service-level Agreements (SLA); and

ISO/IEC 24762:2008, BS25999, ISO 27031, and FINRA Rule 4370

These BC/DR systems must handle data backup and recovery, server backup/recovery, and data center (virtual/physical) backup and recovery. Recovery as a service (RaaS) means that the BC/DR services are being handled by management systems outside the enterprise.

The wide variety of BC/DR requires the security management systems to be able to communicate provisioning, monitor, and control those systems that are being used to back-up and restore data. An interoperable protocol that allows provision and control of data center's data, servers, and data center management devices devices is extremely important to this application. Recovery as a Service (SaaS) indicates that these services need to be able to be remotely management.

The CSA SaaS BC/BR documents indicate how important a standardized I2NSF protocol is.

5.6.1.11. Network Security Devices

Document:
https://downloads.cloudsecurityalliance.org/initiatives/secaas/SecaaS_Cat_10_Network_Security_Implementation_Guidance.pdf

The CSA SaaS Network Security implementation recommendation includes advice on:

How to segment networks,

Network security controls,

Controlling ingress and egress controls such as Firewalls (Stateful), Content Inspection and Control (Network-based), Intrusion Detection System/Intrusion Prevention Systems (IDS/IPS), and Web Application Firewalls,

Secure routing and time,

Denial of Service (DoS) and Distributed Denial of Service (DDoS) Protection/Mitigation,

Virtual Private Network (VPN) with Multiprotocol Label Switching (MPLS) Connectivity (over SSL), Internet Protocol Security (IPsec) VPNs, Virtual Private LAN Service (VPLS), and Ethernet Virtual Private Line (EVPL),

Threat Management,

Forensic Support, and

Privileged User/Use Monitoring.

These network security systems require provisioning, monitoring, and the ability for the security management system to subscribe to

receive logs, snapshots of capture data, and time synchronization. This document states the following:

> "It is critical to understand what monitoring APIs are available from the CSP, and if they match risk and compliance requirements",
>
> "Network security auditors are challenged by the need to track a server and its identity from creation to deletion. Audit tracking is challenging in even the most mature cloud environments, but the challenges are greatly complicated by cloud server sprawl, the situation where the number of cloud servers being created is growing more quickly than a cloud environments ability to manage them."
>
> A valid threat vector for cloud is the API access. Since a majority of CSPs today support public API interfaces available within their networks and likely over the Internet."

The CSA SaaS network security indicates that the I2NSF must be secure so that the I2NSF Client-Agent protocol does not become a valid threat vector. In additions, the need for the management protocol like I2NSF is critical in the sprawl of Cloud environment.

5.6.2. I2NSF Gap Analysis

The CSA Security as a Service (SaaS) document show clearly that there is a gap between the ability of the CSA SaaS devices to have a vendor neutral, inoperable protocol that allow the multiple of network security devices to communicate passing provisioning and informational data. Each of the 10 implementation agreements points to this as a shortage. The I2NSF yang models and protocol is needed according to the CSA SaaS documents.

5.7. In-depth Review of IETF protocols

5.7.1. NETCONF and RESTCONF

The IETF NETCONF working group has developed the basics of the NETCONF protocol focusing on secure configuration and querying operational state. The NETCONF protocol [RFC6241] may be run over TLS [RFC6639] or SSH ([RFC6242]. NETCONF can be expanded to defaults [RFC6243], handling events ([RFC5277] and basic notification [RFC6470], and filtering writes/reads based on network access control models (NACM, [RFC6536]). The NETCONF configuration must be committed to a configuration data store (denoted as config=TRUE). Yang models identify nodes within a configuration data store or an operational data store using a XPath expression (document root ---to --- target source). NETCONF uses an RPC model and provides protocol

for handling configs (get-config, edit-config, copy-config, delete-config, lock, unlock, get) and sessions (close-session, kill-session). The NETCONF Working Group has developed RESTCONF, which is an HTTP-based protocol that provides a programmatic interface for accessing data defined in YANG, using the datastores defined in NETCONF.

RESTCONF supports "two edit condition detections" - time stamp and entity tag. RESTCONF uses a URI encoded path expressions. RESTCONF provides operations to get remote servers options (OPTIONS), retrieve data headers (HEAD), get data (GET), create resource/invoke operation (POST), patch data (PATCH), delete resource (DELETE), or query.

RFCs for NETCONF

- NETCONF [RFC6242]
- NETCONF monitoring [RFC6022]
- NETCONF over SSH [RFC6242]
- NETCONF over TLS [RFC5539]
- NETCONF system notification> [RFC6470]
- NETCONF access-control (NACM) [RFC6536]
- RESTCONF [I-D.ietf-netconf-restconf]
- NETCONF-RESTCONF call home [I-D.ietf-netconf-call-home]
- RESTCONF collection protocol [I-D.ietf-netconf-restconf-collection]
- NETCONF Zero Touch Provisioning [I-D.ietf-netconf-zerotouch]

5.7.2. I2RS Protocol

Based on input from the NETCONF working group, the I2RS working group decided to re-use the NETCONF or RESTCONF protocols and specify additions to these protocols rather than create yet another protocol (YAP).

The required extensions for the I2RS protocol are in the following drafts:

- Ephemeral state [I-D.ietf-i2rs-ephemeral-state],

- Publication-Subscription notifications [I-D.ietf-i2rs-pub-sub-requirements],

- Traceability [I-D.ietf-i2rs-traceability],

- Security requirements [I-D.hares-i2rs-auth-trans]

At this time, NETCONF and RESTCONF cannot handle the ephemeral data store proposed by I2RS, the publication and subscription requirements, the traceability, or the security requirements for the transport protocol and message integrity.

5.7.3. NETMOD Yang modules

NETMOD developed initial Yang models for interfaces [RFC7223]), IP address ([RFC7277]), IPv6 Router advertisement ([RFC7277]), IP Systems ([RFC7317]) with system ID, system time management, DNS resolver, Radius client, SSH, syslog ([I-D.ietf-netmod-syslog-model]), ACLS ([I-D.ietf-netmod-acl-model]), and core routing blocks ([I-D.ietf-netmod-routing-cfg] The routing working group (rtgwg) has begun to examine policy for routing and tunnels.

Protocol specific Working groups have developed yang models for ISIS ([I-D.ietf-isis-yang-isis-cfg]), OSPF ([I-D.ietf-ospf-yang]), and BGP (merge of [I-D.shaikh-idr-bgp-model] and [I-D.zhdankin-idr-bgp-cfg] with the bgp policy proposed multiple Working groups (idr and rtgwg)). BGP Services yang models have been proposed for PPB EVPN ([I-D.tsingh-bess-pbb-evpn-yang-cfg]), EVPN ([I-D.zhuang-bess-evpn-yang]), L3VPN ([I-D.zhuang-bess-l3vpn-yang]), and multicast MPLS/BGP IP VPNs ([I-D.liu-bess-mvpn-yang]).

5.7.4. COPS

One early focus on flow filtering based on policy enforcement of traffic entering a network is the 1990s COPS [RFC2748] design (PEP and PDP) as shown in figure 16. The Policy decision point kept network-wide policy (E.g. ACLs) and sent it to Policy enforcements who then would control what data flows between the two These decision points controlled data flow from PEP to PEP. [RFC3084] describes COPS use for policy provisioning.

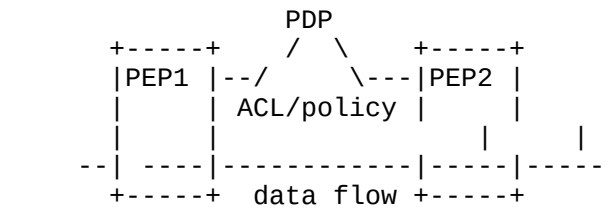


Figure 16

COPS had a design of Policy Enforcement Points (PEP), and policy Decision Points (PDP) as shown in figure 16. These decision points controlled flow from PEP to PEP.

Why COPS is no longer used

Security in the network in 2015 uses specific devices (IDS/IPS, NAT firewall, etc) with specific policies and profiles for each types of device. No common protocol or policy format exists between the policy manager (PDP) and security enforcement points.

COPs RFCs: [RFC4261], [RFC2940], , [RFC3084], , [RFC3483]

Why I2NSF is different COPS

COPS was a protocol for policy related to Quality of Service (QoS) and signalling protocols (e.g. RSVP) (security, flow, and others). I2NSF creates a common protocol between security policy decision points (SPDP) and security enforcement points (SEP). Today's security devices currently only use proprietary protocols. Manufacturers would like a security specific policy enforcement protocol rather than a generic policy protocol.

5.7.5. PCP

As indicated by the name, the Port Control Protocol (PCP) enables an IPv4 or IPv6 host to flexibly manage the IP address and port mapping information on Network Address Translators (NATs) or firewalls, to facilitate communication with remote hosts.

PCP RFCs:

[RFC6887]

[RFC7225]

[I-D.ietf-pcp-authentication]

[I-D.ietf-pcp-optimize-keepalives]

[I-D.ietf-pcp-proxy]

Why is I2NSF different from PCP:

Here are some aspects that I2NSF is different from PCP:

- PCP only supports the management of port and address information rather than any other security functions

- Cover the proxy, firewall and NAT box proposals in I2NSF

5.7.6. NSIS - Next steps in Signalling

NSIS is for standardizing an IP signalling protocol (RSVP) along data path for end points to request its unique QoS characteristics, unique FW policies or NAT needs (RFC5973) that are different from the FW/NAT original setting. The requests are communicated directly to the FW/NAT devices. NSIS is like east-west protocols that require all involved devices to fully comply to make it work.

NSIS is path-coupled, it is possible to message every participating device along a path without having to know its location, or its location relative to other devices (this is particularly a pressing issue when you've got one or more NATs present in the network, or when trying to locate appropriate tunnel endpoints).

A diagram should be added here showing I2NSF and NSIS

Why I2NSF is different than NSIS:

- The I2NSF requests from clients do not go directly to network security devices, but instead to controller or orchestrator that can translate the application/user oriented policies to the involved devices in the interface that they support.

- The I2NSF request does not require all network functions in a path to comply, but it is a protocol between the I2NSF client and the I2NSF Agent in the controller and orchestrator

- I2NSF defines client (applications) oriented descriptors (profiles, or attributes) to request/negotiate/validate the network security functions that are not on the local premises.

Why we belief I2NSF has a higher chance to be deployed than NSIS:

- Open Stack already has a proof-of-concept/preliminary implementation, but the specification is not complete. IETF can play an active role to make the specification for I2NSF is complete. IETF can complete and extend the OpenStack implementation to provide an interoperable specification that can meet the needs and requirements of operators and is workable for suppliers of the technology. The combination of a carefully designed interoperable IETF specification with an open-source code development Open Stack will leverage the strengths of the two communities, and expand the informal ties between the two groups. A software development cycle has the following components: architecture, design specification, coding, and interoperability testing. The IETF can take ownership of the first two steps, and provide expertise and a good working atmosphere (in hack-a-thons) in the last two steps for OpenSTack or other open-source coders.

- IETF has the expertise in security architecture and design for interoperable protocols that span controllers/routers, middle-boxes, and security end-systems.

- IETF has a history of working on interoperable protocols or virtualized network functions (L2VPN, L3VPN) that are deployed by operators in large scale devices. IETF has a strong momentum to create virtualized network functions (see SFC WG in routing) to be deployed in network boxes. [Note: We need to add SACM and others here].

6. Summarized Requirements

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

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

- Policy-driven placement of new function instances in the right administrative domain;

- Attachment of appropriate security and traffic policies to the function instances

- Management of deployed instances in terms of fault monitoring, utilization monitoring, event logging, inventory, etc.

Moreover, an I2NSF must support different deployment scenarios:

- 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.

- 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:

- 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.

- Translation of functional tasks into vendor-specific configuration sets. For example, a firewall policy needs to be converted to vendor-specific configurations.

- 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.

7. IANA Considerations

No IANA considerations exist for this document.

8. Security Considerations

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

- 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.

- 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].

9. Contributors

I2NSF is a group effort. The following people contributed actively to the initial use case text: Diego R. Lopez (Telefonica I+D), Xiaojun Zhuang (China Mobile), Minpeng Qi (China Mobile), Sumandra Majee (F5), Nic Leymann (Deutsche Telekom), Linda Dunbar (Huawei).

10. References

10.1. Normative References

[RFC0791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <http://www.rfc-editor.org/info/rfc791>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <http://www.rfc-editor.org/info/rfc2119>.

10.2. Informative References

[I-D.dunbar-i2rs-discover-traffic-rules] Dunbar, L. and S. Hares, "An Information Model for Filter Rules for Discovery and Traffic for I2RS Filter-Based RIB", draft-dunbar-i2rs-discover-traffic-rules-00 (work in progress), March 2015.

[I-D.hares-i2rs-auth-trans] Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-hares-i2rs-auth-trans-05 (work in progress), August 2015.

[I-D.hares-i2rs-bnp-eca-data-model] Hares, S., Wu, Q., Tantsura, J., and R. White, "An Information Model for Basic Network Policy and Filter Rules", draft-hares-i2rs-bnp-eca-data-model-00 (work in progress), July 2015.

[I-D.hares-i2rs-info-model-service-topo] Hares, S., Wu, W., Wang, Z., and J. You, "An Information model for service topology", draft-hares-i2rs-info-model-service-topo-03 (work in progress), January 2015.

[I-D.ietf-i2rs-architecture]
Atlas, A., Halpern, J., Hares, S., Ward, D., and T. Nadeau, "An Architecture for the Interface to the Routing System", draft-ietf-i2rs-architecture-09 (work in progress), March 2015.

[I-D.ietf-i2rs-ephemeral-state]
Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-02 (work in progress), September 2015.

[I-D.ietf-i2rs-problem-statement]
Atlas, A., Nadeau, T., and D. Ward, "Interface to the Routing System Problem Statement", draft-ietf-i2rs-problem-statement-06 (work in progress), January 2015.

[I-D.ietf-i2rs-pub-sub-requirements]
Voit, E., Clemm, A., and A. Prieto, "Requirements for Subscription to YANG Datastores", draft-ietf-i2rs-pub-sub-requirements-03 (work in progress), October 2015.

[I-D.ietf-i2rs-rib-data-model]
Wang, L., Ananthakrishnan, H., Chen, M., amit.dass@ericsson.com, a., Kini, S., and N. Bahadur, "A YANG Data Model for Routing Information Base (RIB)", draft-ietf-i2rs-rib-data-model-01 (work in progress), September 2015.

[I-D.ietf-i2rs-rib-info-model]
Bahadur, N., Kini, S., and J. Medved, "Routing Information Base Info Model", draft-ietf-i2rs-rib-info-model-07 (work in progress), September 2015.

[I-D.ietf-i2rs-traceability]
Clarke, J., Salgueiro, G., and C. Pignataro, "Interface to the Routing System (I2RS) Traceability: Framework and Information Model", draft-ietf-i2rs-traceability-03 (work in progress), May 2015.

[I-D.ietf-i2rs-usecase-reqs-summary]
Hares, S. and M. Chen, "Summary of I2RS Use Case Requirements", draft-ietf-i2rs-usecase-reqs-summary-01 (work in progress), May 2015.

[I-D.ietf-i2rs-yang-l2-network-topology]
Dong, J. and X. Wei, "A YANG Data Model for Layer-2 Network Topologies", draft-ietf-i2rs-yang-l2-network-topology-01 (work in progress), July 2015.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Tkacik, T., Bahadur, N., and H. Ananthakrishnan, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-01 (work in progress), June 2015.

[I-D.ietf-isis-yang-isis-cfg]
Litkowski, S., Yeung, D., Lindem, A., Zhang, J., and L. Lhotka, "YANG Data Model for ISIS protocol", draft-ietf-isis-yang-isis-cfg-02 (work in progress), March 2015.

[I-D.ietf-netconf-call-home]
Watsen, K., "NETCONF Call Home and RESTCONF Call Home", draft-ietf-netconf-call-home-06 (work in progress), May 2015.

[I-D.ietf-netconf-restconf]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", draft-ietf-netconf-restconf-04 (work in progress), January 2015.

[I-D.ietf-netconf-restconf-collection]
Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Collection Resource", draft-ietf-netconf-restconf-collection-00 (work in progress), January 2015.

[I-D.ietf-netconf-zerotouch]
Watsen, K., Clarke, J., and M. Abrahamsson, "Zero Touch Provisioning for NETCONF Call Home (ZeroTouch)", draft-ietf-netconf-zerotouch-02 (work in progress), March 2015.

[I-D.ietf-netmod-acl-model]
Bogdanovic, D., Sreenivasa, K., Huang, L., and D. Blair, "Network Access Control List (ACL) YANG Data Model", draft-ietf-netmod-acl-model-02 (work in progress), March 2015.

[I-D.ietf-netmod-routing-cfg]
Lhotka, L. and A. Lindem, "A YANG Data Model for Routing Management", draft-ietf-netmod-routing-cfg-19 (work in progress), May 2015.

[I-D.ietf-netmod-syslog-model]
Wildes, C. and K. Sreenivasa, "SYSLOG YANG model", draft-ietf-netmod-syslog-model-03 (work in progress), March 2015.

[I-D.ietf-ospf-yang]
Yeung, D., Qu, Y., Zhang, J., Bogdanovic, D., and K. Sreenivasa, "Yang Data Model for OSPF Protocol", draft-ietf-ospf-yang-00 (work in progress), March 2015.

[I-D.ietf-pcp-authentication]
Wasserman, M., Hartman, S., Zhang, D., and T. Reddy, "Port Control Protocol (PCP) Authentication Mechanism", draft-ietf-pcp-authentication-09 (work in progress), May 2015.

[I-D.ietf-pcp-optimize-keepalives]
Reddy, T., Patil, P., Isomaki, M., and D. Wing, "Optimizing NAT and Firewall Keepalives Using Port Control Protocol (PCP)", draft-ietf-pcp-optimize-keepalives-06 (work in progress), May 2015.

[I-D.ietf-pcp-proxy]
Perreault, S., Boucadair, M., Penno, R., Wing, D., and S. Cheshire, "Port Control Protocol (PCP) Proxy Function", draft-ietf-pcp-proxy-08 (work in progress), May 2015.

[I-D.ietf-sacm-architecture]
Cam-Winget, N., Lorenzin, L., McDonald, I., and l. loxx@cisco.com, "Secure Automation and Continuous Monitoring (SACM) Architecture", draft-ietf-sacm-architecture-03 (work in progress), March 2015.

[I-D.ietf-sacm-terminology]
Waltermire, D., Montville, A., Harrington, D., Cam-Winget, N., Lu, J., Ford, B., and M. Kaeo, "Terminology for Security Assessment", draft-ietf-sacm-terminology-06 (work in progress), February 2015.

[I-D.kini-i2rs-fb-rib-info-model]
Kini, S., Hares, S., Dunbar, L., Ghanwani, A., Krishnan, R., Bogdanovic, D., Tantsura, J., and R. White, "Filter-Based RIB Information Model", draft-kini-i2rs-fb-rib-info-model-01 (work in progress), July 2015.

[I-D.l3vpn-service-yang]
Litkowski, S., Shakir, R., Tomotaki, L., and K. D'Souza, "YANG Data Model for L3VPN service delivery", draft-l3vpn-service-yang-00 (work in progress), February 2015.

[I-D.liu-bess-mvpn-yang]
Liu, Y. and F. Guo, "Yang Data Model for Multicast in MPLS/BGP IP VPNs", draft-liu-bess-mvpn-yang-00 (work in progress), April 2015.

[I-D.shaikh-idr-bgp-model]
Shaikh, A., D'Souza, K., Bansal, D., and R. Shakir, "BGP Model for Service Provider Networks", draft-shaikh-idr-bgp-model-01 (work in progress), March 2015.

[I-D.shaikh-rtgwg-policy-model]
Shaikh, A., Shakir, R., D'Souza, K., and C. Chase, "Routing Policy Configuration Model for Service Provider Networks", draft-shaikh-rtgwg-policy-model-01 (work in progress), July 2015.

[I-D.tsingh-bess-pbb-evpn-yang-cfg]
Tiruveedhula, K., Singh, T., Sajassi, A., Kumar, D., and L. Jalil, "YANG Data Model for PBB EVPN protocol", draft-tsingh-bess-pbb-evpn-yang-cfg-00 (work in progress), March 2015.

[I-D.zhang-i2rs-l1-topo-yang-model]
Zhang, X., Rao, B., and X. Liu, "A YANG Data Model for Layer 1 Network Topology", draft-zhang-i2rs-l1-topo-yang-model-01 (work in progress), March 2015.

[I-D.zhdankin-idr-bgp-cfg]
Alex, A., Patel, K., Clemm, A., Hares, S., Jethanandani, M., and X. Liu, "Yang Data Model for BGP Protocol", draft-zhdankin-idr-bgp-cfg-00 (work in progress), January 2015.

[I-D.zhuang-bess-evpn-yang]
Zhuang, S. and Z. Li, "Yang Model for Ethernet VPN", draft-zhuang-bess-evpn-yang-00 (work in progress), December 2014.

[I-D.zhuang-bess-l3vpn-yang]
Zhuang, S. and Z. Li, "Yang Data Model for BGP/MPLS IP VPNs", draft-zhuang-bess-l3vpn-yang-00 (work in progress), December 2014.

[RFC2748] Durham, D., Ed., Boyle, J., Cohen, R., Herzog, S., Rajan, R., and A. Sastry, "The COPS (Common Open Policy Service) Protocol", RFC 2748, DOI 10.17487/RFC2748, January 2000, <http://www.rfc-editor.org/info/rfc2748>.

[RFC2940] Smith, A., Partain, D., and J. Seligson, "Definitions of Managed Objects for Common Open Policy Service (COPS) Protocol Clients", RFC 2940, DOI 10.17487/RFC2940, October 2000, <http://www.rfc-editor.org/info/rfc2940>.

[RFC3084] Chan, K., Seligson, J., Durham, D., Gai, S., McCloghrie, K., Herzog, S., Reichmeyer, F., Yavatkar, R., and A. Smith, "COPS Usage for Policy Provisioning (COPS-PR)", RFC 3084, DOI 10.17487/RFC3084, March 2001, <http://www.rfc-editor.org/info/rfc3084>.

[RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, DOI 10.17487/RFC3303, August 2002, <http://www.rfc-editor.org/info/rfc3303>.

[RFC3304] Swale, R., Mart, P., Sijben, P., Brim, S., and M. Shore, "Middlebox Communications (midcom) Protocol Requirements", RFC 3304, DOI 10.17487/RFC3304, August 2002, <http://www.rfc-editor.org/info/rfc3304>.

[RFC3483] Rawlins, D., Kulkarni, A., Bokaemper, M., and K. Chan, "Framework for Policy Usage Feedback for Common Open Policy Service with Policy Provisioning (COPS-PR)", RFC 3483, DOI 10.17487/RFC3483, March 2003, <http://www.rfc-editor.org/info/rfc3483>.

[RFC3484] Draves, R., "Default Address Selection for Internet Protocol version 6 (IPv6)", RFC 3484, DOI 10.17487/RFC3484, February 2003, <http://www.rfc-editor.org/info/rfc3484>.

[RFC4080] Hancock, R., Karagiannis, G., Loughney, J., and S. Van den Bosch, "Next Steps in Signaling (NSIS): Framework", RFC 4080, DOI 10.17487/RFC4080, June 2005, <http://www.rfc-editor.org/info/rfc4080>.

[RFC4261] Walker, J. and A. Kulkarni, Ed., "Common Open Policy Service (COPS) Over Transport Layer Security (TLS)", RFC 4261, DOI 10.17487/RFC4261, December 2005, <http://www.rfc-editor.org/info/rfc4261>.

[RFC4949] Shirey, R., "Internet Security Glossary, Version 2", FYI 36, RFC 4949, DOI 10.17487/RFC4949, August 2007, <http://www.rfc-editor.org/info/rfc4949>.

[RFC5189] Stiemerling, M., Quittek, J., and T. Taylor, "Middlebox Communication (MIDCOM) Protocol Semantics", RFC 5189, DOI 10.17487/RFC5189, March 2008, <http://www.rfc-editor.org/info/rfc5189>.

[RFC5277] Chisholm, S. and H. Trevino, "NETCONF Event Notifications", RFC 5277, DOI 10.17487/RFC5277, July 2008, <http://www.rfc-editor.org/info/rfc5277>.

[RFC5539] Badra, M., "NETCONF over Transport Layer Security (TLS)", RFC 5539, DOI 10.17487/RFC5539, May 2009, <http://www.rfc-editor.org/info/rfc5539>.

[RFC5973] Stiemerling, M., Tschofenig, H., Aoun, C., and E. Davies, "NAT/Firewall NSIS Signaling Layer Protocol (NSLP)", RFC 5973, DOI 10.17487/RFC5973, October 2010, <http://www.rfc-editor.org/info/rfc5973>.

[RFC6022] Scott, M. and M. Bjorklund, "YANG Module for NETCONF Monitoring", RFC 6022, DOI 10.17487/RFC6022, October 2010, <http://www.rfc-editor.org/info/rfc6022>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <http://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <http://www.rfc-editor.org/info/rfc6242>.

[RFC6243] Bierman, A. and B. Lengyel, "With-defaults Capability for NETCONF", RFC 6243, DOI 10.17487/RFC6243, June 2011, <http://www.rfc-editor.org/info/rfc6243>.

[RFC6436] Amante, S., Carpenter, B., and S. Jiang, "Rationale for Update to the IPv6 Flow Label Specification", RFC 6436, DOI 10.17487/RFC6436, November 2011, <http://www.rfc-editor.org/info/rfc6436>.

[RFC6470] Bierman, A., "Network Configuration Protocol (NETCONF) Base Notifications", RFC 6470, DOI 10.17487/RFC6470, February 2012, <http://www.rfc-editor.org/info/rfc6470>.

[RFC6536] Bierman, A. and M. Bjorklund, "Network Configuration Protocol (NETCONF) Access Control Model", RFC 6536, DOI 10.17487/RFC6536, March 2012, <http://www.rfc-editor.org/info/rfc6536>.

[RFC6639] King, D., Ed. and M. Venkatesan, Ed., "Multiprotocol Label Switching Transport Profile (MPLS-TP) MIB-Based Management Overview", RFC 6639, DOI 10.17487/RFC6639, June 2012, <http://www.rfc-editor.org/info/rfc6639>.

[RFC6887] Wing, D., Ed., Cheshire, S., Boucadair, M., Penno, R., and P. Selkirk, "Port Control Protocol (PCP)", RFC 6887, DOI 10.17487/RFC6887, April 2013, <http://www.rfc-editor.org/info/rfc6887>.

[RFC7223] Bjorklund, M., "A YANG Data Model for Interface Management", RFC 7223, DOI 10.17487/RFC7223, May 2014, <http://www.rfc-editor.org/info/rfc7223>.

[RFC7225] Boucadair, M., "Discovering NAT64 IPv6 Prefixes Using the Port Control Protocol (PCP)", RFC 7225, DOI 10.17487/RFC7225, May 2014, <http://www.rfc-editor.org/info/rfc7225>.

[RFC7277] Bjorklund, M., "A YANG Data Model for IP Management", RFC 7277, DOI 10.17487/RFC7277, June 2014, <http://www.rfc-editor.org/info/rfc7277>.

[RFC7317] Bierman, A. and M. Bjorklund, "A YANG Data Model for System Management", RFC 7317, DOI 10.17487/RFC7317, August 2014, <http://www.rfc-editor.org/info/rfc7317>.

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA

Email: shares@ndzh.com

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

Email: antonio.pastorperales@telefonica.com

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

Email: wangkeyj@chinamobile.com

Dacheng Zhang
Beijing
China

Email: dacheng.zdc@aliabab-inc.com

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

Email: myo.zarny@gs.com