I2NSF WG
Internet-Draft
Intended status: Informational
Expires: September 7, 2017

S. Hares
R. Moskowitz
Huawei
D. Zhang
March 6, 2017

Analysis of Existing work for I2NSF
draft-ietf-i2nsf-gap-analysis-03.txt

Abstract

This document analyzes the current state of the art for security management devices and security devices technologies in industries and the existing IETF work/protocols that are 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.

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 September 7, 2017.

Table of Contents

1. Introduction

This documents provides a gap analysis for I2NSF.

1.1. What is I2NSF

A Network Security Function (NSF) ensures integrity, confidentiality and availability of network communications, detects unwanted activity, and/or blocks out or at least mitigates the effects of unwanted activity. NSFs are provided and consumed in increasingly diverse environments. For example, users of NSFs could consume network security services offered on multiple security products hosted one or more service provider,their own enterprises, or a combination of the two.

The lack of standard interfaces to control and monitor the behavior of NSFs makes it virtually impossible for security service providers to automate service offerings that utilize different security functions from multiple vendors.

The Interface to Network Service Functions (I2NSF) work proposes to standardize a set of software interfaces 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 is 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. The Service Layer is out of scope for this phase of I2NSF's work.

For the I2NSF Capability Layer, the I2NSF work proposes an interoperable protocol that passes NSF provisioning rules and orchestration information between the I2NSF client on a network manager and the I2NSF agent on an NSF. 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).

1.2. Structure of this Document

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

IETF working groups (Section 2)

ETSI Network Functions Virtualization Industry Specification Group (ETSI NFV ISG), (Section 3)

OPNFV Open Source Group (Section 4)

Open Stack - Firewall as a service (OpenStack Firewall FaaS) (Section 5) (http://docs.openstack.org/admin-guide-cloud/content/install_neutron-fwaas-agent.html)

Cloud Security Alliance Security (CSA)as a Service (Section 6) (https://cloudsecurityalliance.org/research/secaas/#_overview)

In-Depth Review of Some IETF Protocols (Section 7)

1.3. Terms and Definitions

1.3.1. Requirements Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119, BCP 14 [RFC2119] and indicate requirement levels for compliant CoAP.

1.3.2. Definitions

The following are a few definitions out of the terminology draft utilized in this draft. For additional definitions please see: [I-D.hares-i2nsf-terminology].

Network Security Function (NSF): is a function that is provided as a set of security-related service function. Typically, an NSF may be responsible for detecting unwanted activity and blocking/ mitigating the effect of such unwanted activity in order to fulfil the service requirements. The NSF can help in supporting communication stream integrity and confidentiality.

Cloud Data Center (DC): A data center that may/may not be run on the 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) focuses 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 Functions (NSFs) that may be hosted and managed by service providers or a different administrative entity.

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.

I2NSF server/agent: A software instance that implements a network security function that receives provisioning information and requests operational data (e.g. monitoring data) from an I2NSF client. It is also responsible for enforcing the policies that it receives from an I2NSF client.

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)

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.

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.

2.1. Traffic Filters

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]) - for supporting signaling about a data flow along its path, and

o Port Control Protocol (PCP) - allows the IPv4/IPv6 host to control how IPv6/IPv4 packets are translated and forwarded by 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.

Route filtering is outside the scope of the flow filtering, but the flow filtering may be impacted by route filtering. An initial model for routing policy is in [I-D.ietf-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 Section 7.

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 list filters which can permit or deny traffic flow based on headers at the MAC Layer, 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.

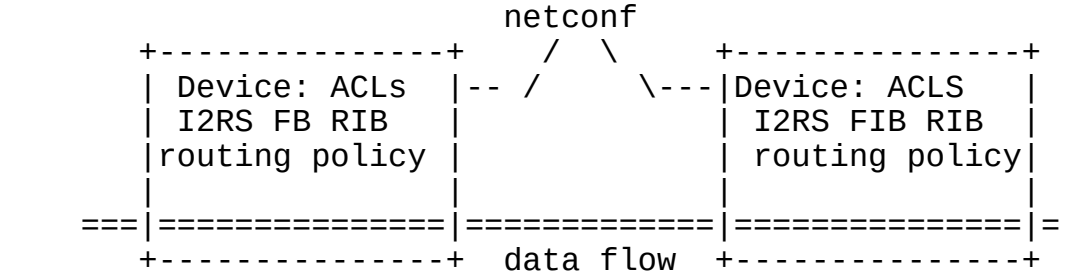


Figure 1

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]). See Section 7.2 on I2RS for additional details on I2RS.

The vocabulary of the [I-D.ietf-netmod-acl-model] is limited, so additional protocol independent filters has been written for the I2RS Filter-Based RIBs in [I-D.hares-i2rs-pkt-eca-data-model].

One thing important to note is that NETCONF and RESTCONF manage device layer YANG models. However, as Figure 2 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).

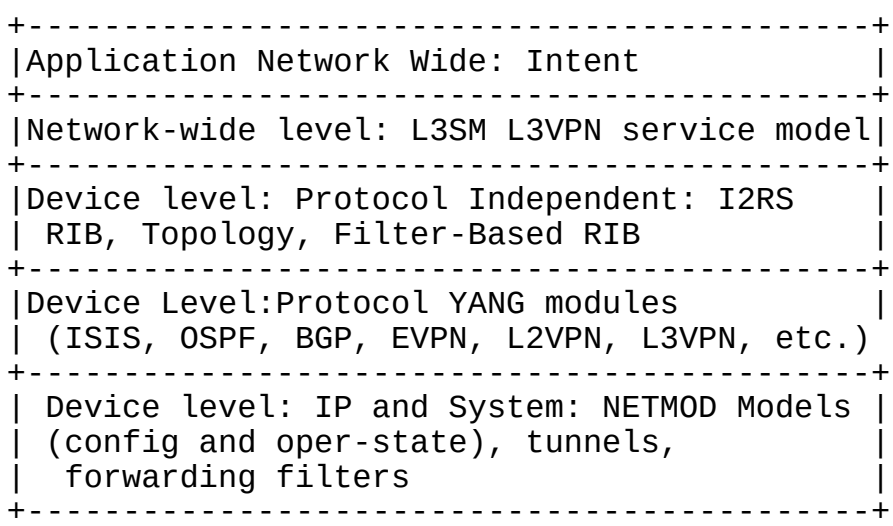


Figure 2 Levels of YANG modules

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 the requisite security expertise for such NSFs (IDP/IPS, Web-filter, security-based deep-packet inspection, etc.) 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.

2.1.2. Middle-box Filters

2.1.2.1. Midcom

Midcom Summary: MIDCOM developed the protocols for applications to communicate with middle boxes. However, MIDCOM have not been used by the industry for a long time. A main reason is that MIDCOM had a lot of IPR encumbered technology and IPR was likely a bigger problem for IETF at that time 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 words, there is a trust model in MIDCOM 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 from Midcom

MIDCOM is different from I2NSF because its SNMP scheme does not work with the virtual network security functions (vNSF) management.

MidCom RFCs:

[RFC3303] - Midcom architecture

[RFC5189] - Midcom Protocol Semantics

[RFC3304] - Midcom protocol requirements

2.1.3. Security Work

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.

Defining the requirements for an I2NSF protocol to pass this traffic. (Ideally by 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 Continuous Monitoring), DOTS (DDoS Open Threat Signalling), and MILE (Management Incident Lightweight Exchange).

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 7.4.

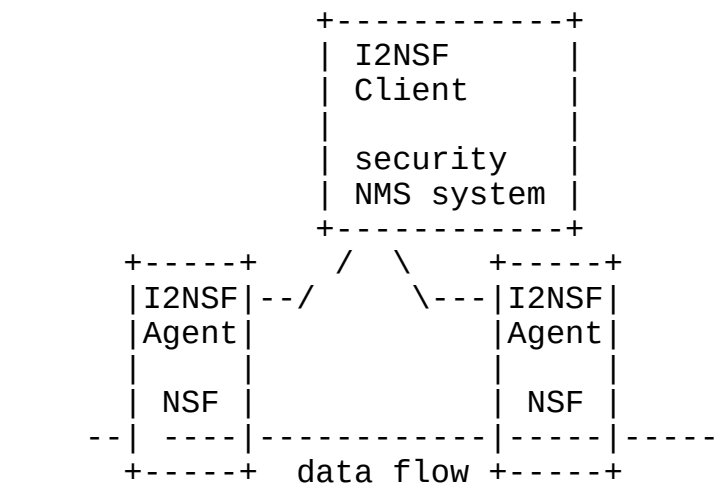


Figure 2

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

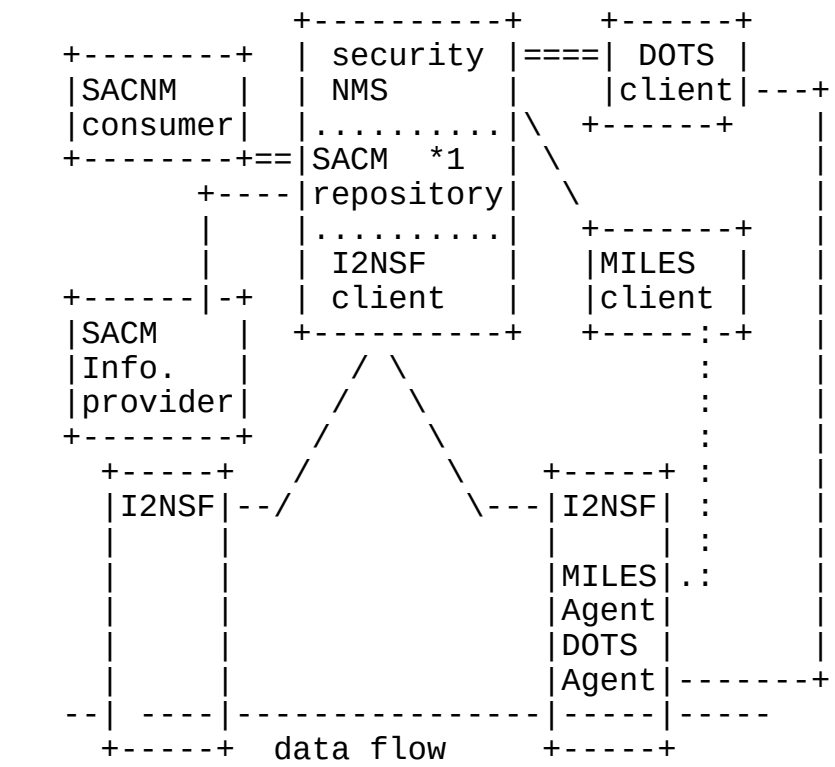
- SACM [I-D.ietf-sacm-architecture] describes an architecture which tries to determine if the end-point security policies and the reality (denoted as security posture) align. [I-D.ietf-sacm-terminology] defines posture as the configuration and/or status of hardware or software on an endpoint as it pertains 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.

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 3

Figure 3 provides a diagram of a system in which the I2NSF, SACM, DOTS and MILE 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 providers and SACM consumers. 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 MILE client on a security network management system talking to the MILE 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.

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 ACLs defined NETMOD, the DOTS black-lists are not sufficient for all filters or control desired by the NSF boxes.

The incident data captured by MILE 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.

3. ETSI NFV

3.1. ETSI Overview

Network Functions Virtualization (NFV) provides 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. However, the exterior network beyond the service provider NSFs or its customer's NSFs is becoming extremely narrow as NSFs are becoming more pervasive in any portion of networks (service providers, customers, or access networks).

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 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 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 framework is complementary to the NFV and other security frameworks. The I2NSF management protocol will be deployed in networks to provide a common management protocol to manage NSF software/devices whether the devices are physical or virtual. The

ETSI NFV security is also deployed along-side other security functions (AAA, SACM, DOTS, or MILE devices) and deep-packet stateful inspections.

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 4 shows. The ETSI 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.

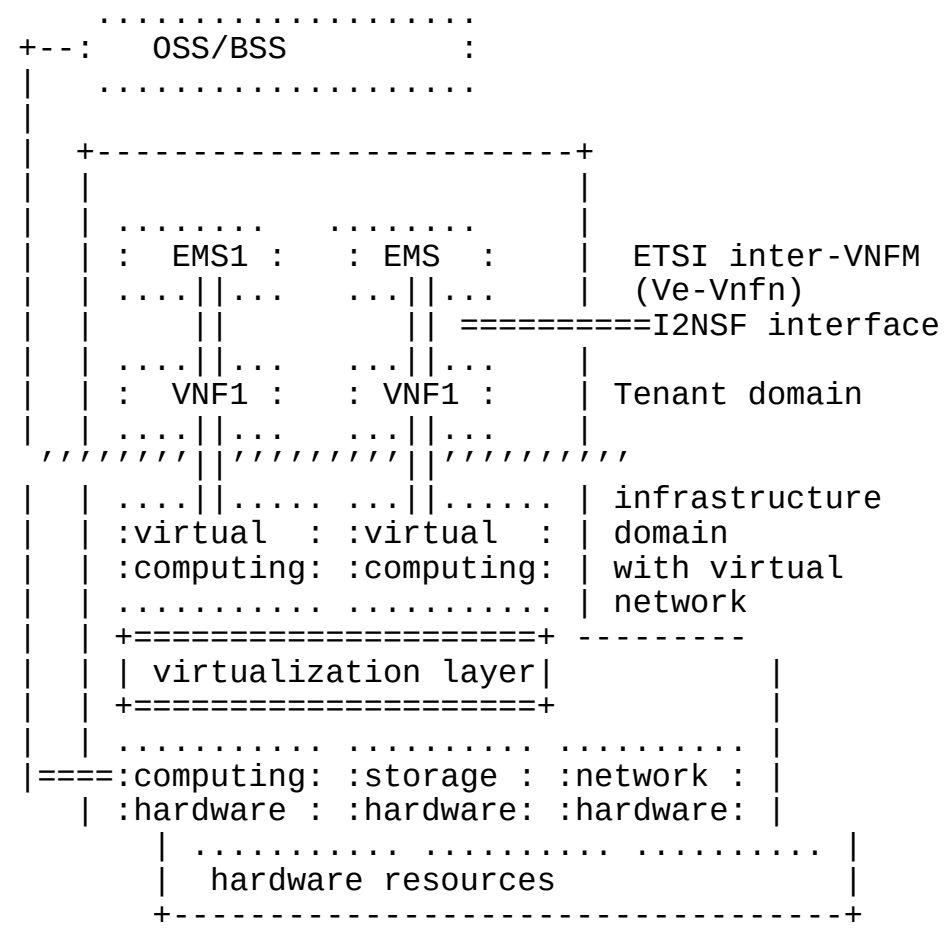


Figure 4

The ETSI proof-of-concept demonstrations have been done for the security proof of concepts:

- #16 - NFVIaaS with Secure, SDN controlled WAN Gateway,

3.2. I2NSF Gap Analysis

The I2NSF protocol/interface can be deployed for security devices along-side the network/host configuration done by NETCONF/RESTCONF or the routing system interface provided by I2RS that handles.

In the current NFV-related architecture, there is no interoperable protocol defined between a security manager and various NSF devices to provision security functions. The result is that service providers have to manage the interoperability security manager and different NSF devices using proprietary 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 on which providers may start their NFV development work. However, this code base faces the same problem. There is no defacto standard protocol.

4. OPNFV

The OPNFV (www.opnfv.org) is a carrier-grade integrated, open source platform focused on accelerating the introduction of new Network Functions 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.

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 carte" to enforcement related security managers. A security management system integrates mechanisms of different security aspects. This project intends 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 5 shows. This figure 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 compute and Swift for object storage. Each system has a configuration file and its own security policy. The system lacks a synchronization mechanism to build a complete secure configuration for OPNFV.

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

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

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

Moon addresses these issues adding authorization, logging, IDS, enforcement of network policy, and storage protection. Moon release C (2016) plans to:

- Define an identity federation scenario between OpenStack and OpenDaylight,

- Implement an authentication driver in ODL to delegate authentication to OpenStack/Keystone,

- Implement a command line tool for administration and testing,

- Implement a graphic interface for identity management for both OpenStack and OpenDaylight,

- Set up identity federation testbed,

- Define identity federation scenarios with other SDN controllers, and

- Define authorization federation scenarios with OpenDaylight.

Deliverable time frame: Moon Release 3 (mid-year 2016)

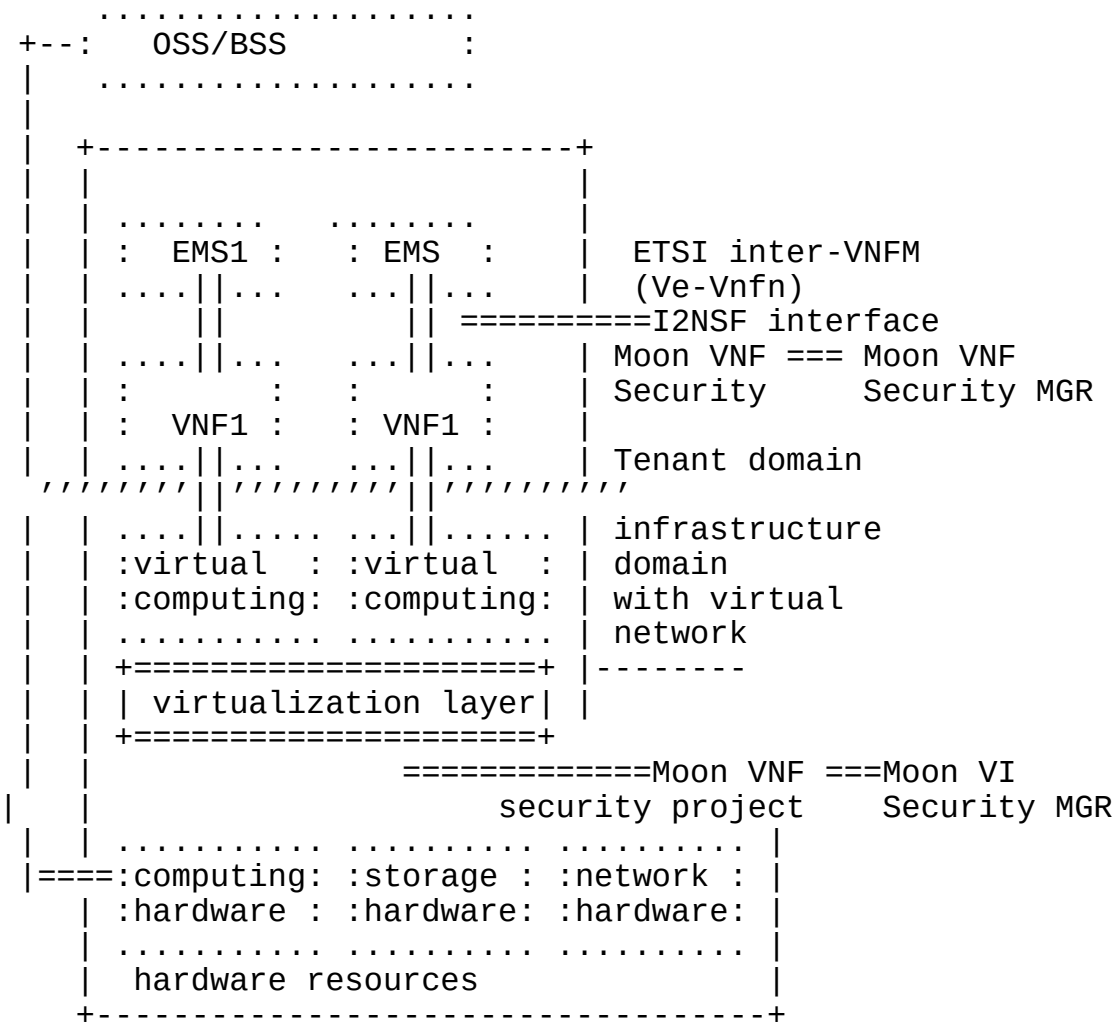


Figure 5

4.2. Gap Analysis for OPNFV Moon Project

OpenStack Congress does not provide vendor independent systems.

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

5.1. Overview of API for Security Group

The security group rules provide ingress and egress traffic filters based on port. The default rule for the group policy drops all ingress traffic and allows all egress traffic. The group policy allows users to add additional groups with additional filters that change the default behaviour. To utilize the security groups, the networking plug-in for OpenStack must implement the security group API. The following plug-ins in OpenStack 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.

5.2. Overview of Firewall 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, or 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

5.3. 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. 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 need for a single management interfaces for a variety of security devices.

Phase 1 of I2NSF is proposes rules that will include Event-Condition-Action matches (ECA) rules with:

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

context based matches on schedule state and schedule.

basic ations of deny, permit, and mirror,

advanced actions of: IPS signature filtering and URL filtering.

[Editorial note: do we need more matches or actions?]

6. CSA Secure Cloud

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

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 guidelines.

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 problem. This 2012 implementation report confirms there is a gap with IAM.

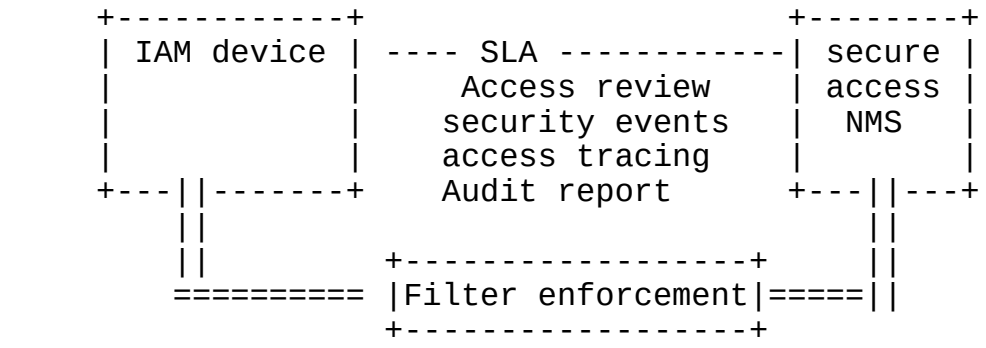


Figure 6

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

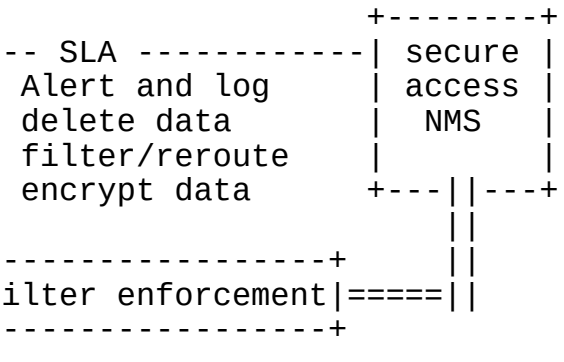


Figure 7

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.

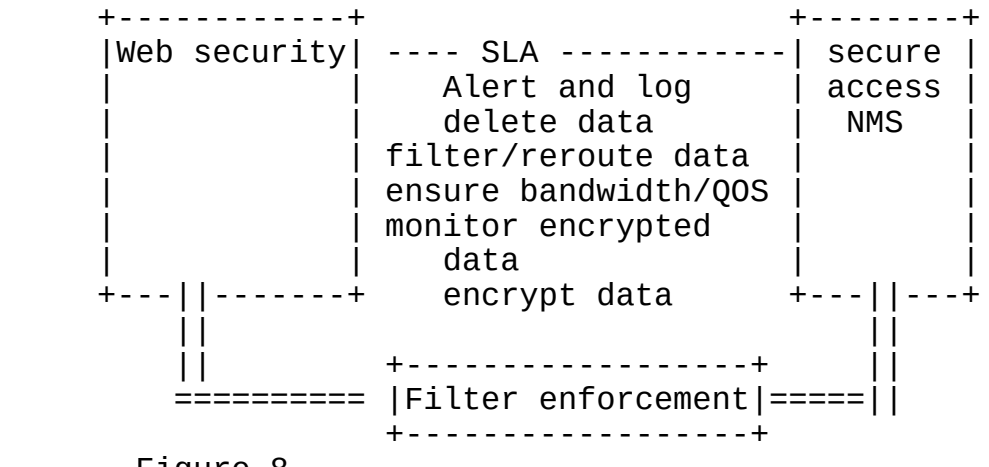


Figure 8

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.

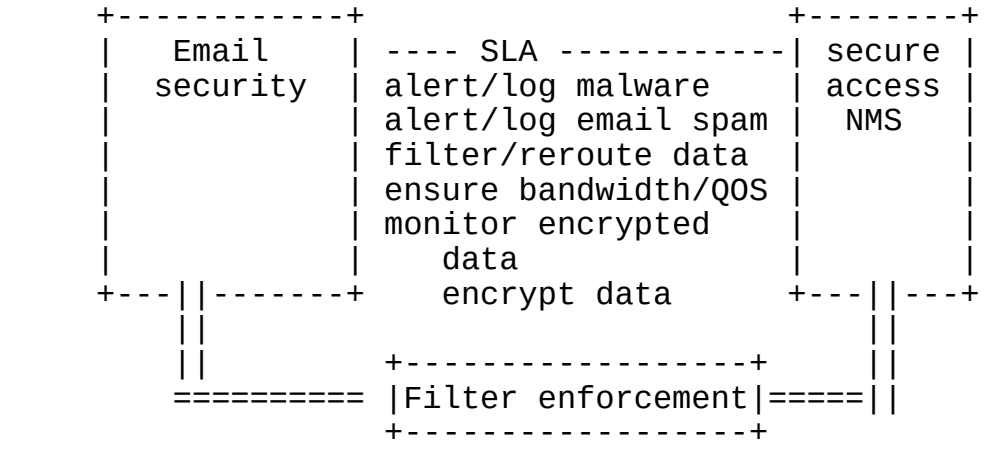


Figure 9

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,

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

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:

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

Intrusion response includes both:

- Automatic, Manual, or Hybrid Mechanisms,
- 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 or intrusion protection devices. Management of these systems 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.

In order to be able performing these management actions on NSF devices from different vendors, the intrusion security management systems need a standard mangement protocol that all the NSF vendors support.

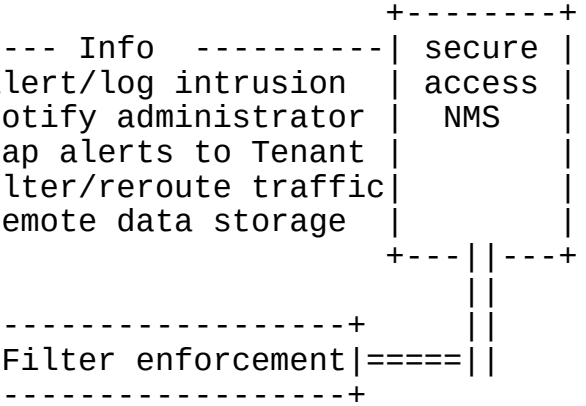


Figure 10

The I2NSF manager - I2NSF (server/agent) protocol is designed to fill this gap.

6.1.8. Security Information and Event Management(SIEM)

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

The Security Information and Event Management (SIEM) 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 SIEM 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 a prime candidate to have an 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.

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, Identity-based Encryption, and others;

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

Cryptographic Security Standards and Guidelines

Encryption services typically require that 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.

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 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;
- Cold 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.

BC/DR requires 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.

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.

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 shortcoming. Standard I2NSF YANG models and an I2NSF protocol are needed according to the CSA SaaS documents.

7. IEEE security

7.1. Port-based Network Access Control [802.1X]

802.1x defines encapsulation of Extensible Authentication Protocol (EAP) [RFC3748] over IEEE 802 (EAP over LAN, or EAPOL). It is widely deployed on both wired and Wi-Fi Networks.

EAP provides support for security from passwords to challenge-response tokens and public-key infrastructure certificates.

802.1 has three concepts:

- the supplicant - the user or client who wants to be authenticated
- authentication server - the server doing the authrentication (e.g. radius server), and
- the authenticator - the device in-between authentication server and supplicant (e.g. wireless Access Point (AP).

A normal sequence is below:

```
supplicant     authenticator          authentication server
==========     ==================     ========================
       <---- EAP-Request/Identity

EAP-Response/Identity---->
                        EAP-Response/Identity---gt;
                                                  <---------Challenge
             <------Challenge

Challenge
  response --------->
                      Challenge
                                            Response --------->
```

Gap:

This basic service provides access, but today’s access use cases are more complex. IEEE 801.X has ben attched using the Man-in-the-middle attacks. Another weakness of 802.1X is the speed of the EAP protocols processing with the radius server.

Note: Editor - more is needed here

7.2. MAC security (802.1AE)

MACsec security secures a link rather than a conversation for 802.1 LANs (VLANs 802.1Q, Provider Bridges 802.1AD). MACsec counters the 802.1X man-in the middle attacks.

MACsec (in 802.1AE) provides MAC-layer encryption over wired networks by using out-of-band methods for encryption keying. The MACsec Key Agreement (MKA) Protocol provides the required session keys and manages the required encryption keys. MKA and MACsec are implemented after successful authentication using the 802.1x Extensible Authentication Protocol (EAP) framework. Only hosts link which face

the network can be secured with MACSec. These links connect the host to the network access devices.

Switch using MACsec accepts either MACsec or non-MACsec frames based on policy set. The NSF controller can set within the switches configuration whether MACSec frames are accepted. Accepted MACsec frames are encrypted and protected with an integrity check value (ICV). The switch after receiving frames from the client, decrypts them and calculates the correct ICV by using session keys provided by MKA. The switch compares that ICV to the ICV within the frame. If they are not identical, the frame is dropped. The switch also encrypts and adds an ICV to any frames sent over the secured port (the access point used to provide the secure MAC service to a client) using the current session key.

The MKA Protocol manages the encryption keys used by the underlying MACsec protocol. The basic requirements of MKA are defined in 802.1x-REV. The MKA Protocol extends 802.1x to allow peer discovery with confirmation of mutual authentication and sharing of MACsec secret keys to protect data exchanged by the peers. MKA protocol ues EAP-over-LAN (EAPOL) packet. EAP authentication produces a master session key (MSK) shared by both partners in the data exchange. Entering the EAP session ID generates a secure connectivity association key name (CKN). Because the switch is the authenticator, and the key serer, it can generating a random 128-bit secure association key (SAK), which it sends it to the client partner. The client is never a key server and can only interact with a single MKA entity, the key server. After key derivation and generati

Gap Analysis:

I2NSF Devices must handle the existence of MSEC within the network.

7.3. Secure Device Identity [802.1AR]

802.1AR does the following:

Supports trail of trust from manufacturer to user, and

Defines how a Secure Device Identifier (DevId) may be cryptographically bound to a device to support device identity authentication.

DevIDs are composed of a secure device identifier secret and a secure device identifier credential. These IDs can be controlled by the product manufacturer (IDevID, an initially installed identity) or by the end-user (LDevID, a subsequent locally

significant identity derived from the IDevID). DevIDs cannot be be changed by the end-user.

One attack mitigation can be to disable support for DeVIDs or limit to know DeVIDs.

GAP analysis:

I2NSF controllers need to support 802.1AR device management.

8. In-depth Review of IETF protocols

8.1. 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 data stores defined in NETCONF.

RESTCONF supports "two edit condition detections" - time stamp and entity tag. RESTCONF uses 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]

8.2. 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:

- [I-D.ietf-i2rs-ephemeral-state],
- [I-D.ietf-i2rs-pub-sub-requirements] (Publication-Subscription notifications,
- [I-D.ietf-i2rs-traceability]
- [I-D.ietf-i2rs-protocol-security-requirements]

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.

8.3. 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 ([I-D.ietf-idr-bgp-model].

BGP Services YANG models have been proposed for

- EVPN [I-D.brissette-bess-evpn-yang],
- L2VPN [I-D.shah-bess-l2vpn-yang],
- L3VPN [I-D.li-bess-l3vpn-yang] and [I-D.hu-bess-l2vpn-service-yang],

TEAS working group has proposed [I-D.ietf-teas-yang-te-topo], and [I-D.ietf-teas-yang-rsvp].

MPLS/PCE/CCAMP groups have proposed the following Yang modules:

- [I-D.raza-mpls-ldp-mldp-yang]
- [I-D.saad-mpls-static-yang],
- [I-D.zheng-mpls-lsp-ping-yang-cfg],
- [I-D.pkd-pce-pcep-yang], and
- [I-D.zhang-ccamp-transport-ctrlnorth-yang].

8.4. 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 11. The COPS policy decision points (PDP) managed network-wide policy (e.g. ACLs) by installing this policy in policy enforcement points (PEPs) on the edge of the network. These PEPs had firewall-like functions that control what data flows into the network at a PEP point, and data flow out of a network at a PEP. [RFC3084] describes COPS usages for policy provisioning.

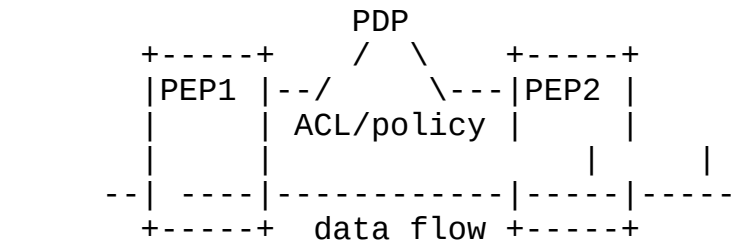


Figure 11

Why COPS is no longer used

Network security today 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], and [RFC3483].

Why I2NSF is Different from COPS

COPS was a protocol for policy related to Quality of Service (QoS) and signaling 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.

8.5. 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:

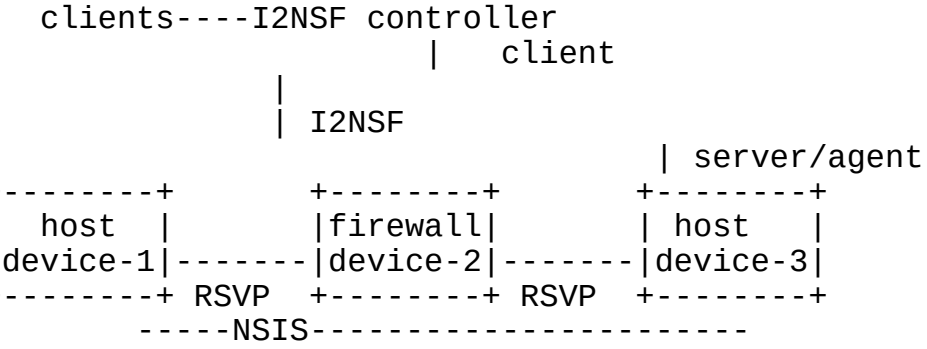
Here are some aspects that I2NSF is different from PCP:

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

8.6. NSIS - Next Steps in Signaling

NSIS aims to standardize an IP signaling protocol (RSVP) along the data path for end points to request their unique QoS characteristics, unique FW policies or NAT needs (RFC5973) that are different from the FW/NAT original settings. 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 one or more NATs present in the network, or when trying to locate appropriate tunnel endpoints).



Why I2NSF is Different from NSIS:

- 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/Server

- 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 I2NSF may have higher chance to be deployed than NSIS:

- OpenStack 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 OpenStack 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.

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

- o 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].

9. IANA Considerations

No IANA considerations exist for this document.

10. Security Considerations

No security considerations are involved with a gap analysis.

11. Contributors

The following people have contributed to this document: Hosnieh Rafiee, and Myo Zarny. Myo Zarny provided the authors with extensive comments, great suggestions, and valuable insights on alternative views.

12. References

12.1. 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>.

12.2. Informative References

[I-D.brissette-bess-evpn-yang]
Brissette, P., Shah, H., Li, Z., Tiruveedhula, K., Singh, T., and I. Hussain, "Yang Data Model for EVPN", draft-brissette-bess-evpn-yang-01 (work in progress), December 2015.

[I-D.hares-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., and L. Xia, "I2NSF Terminology", draft-hares-i2nsf-terminology-00 (work in progress), March 2016.

[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.hares-i2rs-pkt-eca-data-model]
Hares, S., Wu, Q., and R. White, "Filter-Based Packet Forwarding ECA Policy", draft-hares-i2rs-pkt-eca-data-model-02 (work in progress), February 2016.

[I-D.hu-bess-l2vpn-service-yang]
hu, f., Chen, R., and J. Yao, "L2VPN Service YANG Model", draft-hu-bess-l2vpn-service-yang-00 (work in progress), March 2016.

[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-11 (work in progress), December 2015.

[I-D.ietf-i2rs-ephemeral-state]
Haas, J. and S. Hares, "I2RS Ephemeral State Requirements", draft-ietf-i2rs-ephemeral-state-23 (work in progress), November 2016.

[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-11 (work in progress), May 2016.

[I-D.ietf-i2rs-protocol-security-requirements]
Hares, S., Migault, D., and J. Halpern, "I2RS Security Related Requirements", draft-ietf-i2rs-protocol-security-requirements-17 (work in progress), September 2016.

[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-09 (work in progress), May 2016.

[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-07 (work in progress), January 2017.

[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-10 (work in progress), December 2016.

[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-11 (work in progress), May 2016.

[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-03 (work in progress), July 2016.

[I-D.ietf-i2rs-yang-network-topo]
Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A Data Model for Network Topologies", draft-ietf-i2rs-yang-network-topo-12 (work in progress), March 2017.

[I-D.ietf-idr-bgp-model]
Shaikh, A., Shakir, R., Patel, K., Hares, S., D'Souza, K., Bansal, D., Clemm, A., Zhdankin, A., Jethanandani, M., and X. Liu, "BGP Model for Service Provider Networks", draft-ietf-idr-bgp-model-01 (work in progress), January 2016.

[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-l3sm-l3vpn-service-model]
Litkowski, S., Shakir, R., Tomotaki, L., Ogaki, K., and K. D'Souza, "YANG Data Model for L3VPN service delivery", draft-ietf-l3sm-l3vpn-service-model-05 (work in progress), March 2016.

[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-rtgwg-policy-model]
Shaikh, A., Shakir, R., D'Souza, K., and C. Chase, "Routing Policy Configuration Model for Service Provider Networks", draft-ietf-rtgwg-policy-model-00 (work in progress), September 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-05 (work in progress), October 2015.

[I-D.ietf-sacm-terminology]
Birkholz, H., Lu, J., and N. Cam-Winget, "Secure Automation and Continuous Monitoring (SACM) Terminology", draft-ietf-sacm-terminology-09 (work in progress), March 2016.

[I-D.ietf-teas-yang-rsvp]
Beeram, V., Saad, T., Gandhi, R., Liu, X., Shah, H., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for Resource Reservation Protocol (RSVP)", draft-ietf-teas-yang-rsvp-06 (work in progress), October 2016.

[I-D.ietf-teas-yang-te-topo]
Liu, X., Bryskin, I., Beeram, V., Saad, T., Shah, H., and O. Dios, "YANG Data Model for TE Topologies", draft-ietf-teas-yang-te-topo-06 (work in progress), October 2016.

[I-D.kini-i2rs-fb-rib-info-model]
Kini, S., Hares, S., Dunbar, L., Ghanwani, A., Krishnan, R., Bogdanovic, D., and R. White, "Filter-Based RIB Information Model", draft-kini-i2rs-fb-rib-info-model-03 (work in progress), February 2016.

[I-D.li-bess-l3vpn-yang]
Li, Z., Zhuang, S., Liu, X., Haas, J., Esale, S., and B. Wen, "Yang Data Model for BGP/MPLS IP VPN", draft-li-bess-l3vpn-yang-00 (work in progress), October 2015.

[I-D.pkd-pce-pcep-yang]
Dhody, D., Hardwick, J., Beeram, V., and j. jefftant@gmail.com, "A YANG Data Model for Path Computation Element Communications Protocol (PCEP)", draft-pkd-pce-pcep-yang-06 (work in progress), July 2016.

[I-D.raza-mpls-ldp-mldp-yang]
Raza, K., Asati, R., Liu, X., Esale, S., Chen, X., and H. Shah, "YANG Data Model for MPLS LDP and mLDP", draft-raza-mpls-ldp-mldp-yang-04 (work in progress), July 2016.

[I-D.saad-mpls-static-yang]
Saad, T., Raza, K., Gandhi, R., Liu, X., Beeram, V., Shah, H., Bryskin, I., Chen, X., Jones, R., and B. Wen, "A YANG Data Model for MPLS Static LSPs", draft-saad-mpls-static-yang-03 (work in progress), May 2016.

[I-D.shah-bess-l2vpn-yang]
Shah, H., Brissette, P., Rahman, R., Raza, K., Li, Z., Zhuang, S., Wang, H., Chen, I., Ahmed, S., Bocci, M., Hardwick, J., Esale, S., Tiruveedhula, K., tsingh@juniper.net, t., Hussain, I., Wen, B., Walker, J., Delregno, N., Jalil, L., and M. Joecylyn, "YANG Data Model for MPLS-based L2VPN", draft-shah-bess-l2vpn-yang-01 (work in progress), March 2016.

[I-D.zhang-ccamp-transport-ctrlnorth-yang]
Zhang, X., Jing, R., Jian, W., Ryoo, J., Xu, Y., and D. King, "YANG Models for the Northbound Interface of a Transport Network Controller: Requirements, Functions, and a List of YANG Models", draft-zhang-ccamp-transport-ctrlnorth-yang-00 (work in progress), March 2016.

[I-D.zheng-mpls-lsp-ping-yang-cfg]
Zheng, L., Aldrin, S., Zheng, G., Mirsky, G., and R. Rahman, "Yang Data Model for LSP-PING", draft-zheng-mpls-lsp-ping-yang-cfg-04 (work in progress), October 2016.

[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>.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, Ed., "Extensible Authentication Protocol (EAP)", RFC 3748, DOI 10.17487/RFC3748, June 2004, <http://www.rfc-editor.org/info/rfc3748>.

[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

Bob Moskowitz
Huawei
Oak Park, MI 48237

Email: rgm@labs.htt-consult.com

Dacheng Zhang
Beijing
China

Email: dacheng.zdc@aliabab-inc.com