Network Working Group                                    L. Dunbar
Internet Draft                                              Huawei
Intended status: Informational                            M. Zarny
Expires: November 2015                              Goldman Sachs
                                                      C. Jacquenet
                                                       M. Boucadair
                                                    France Telecom
                                                    S. Chakrabarty
                                                          US Ignite

                                                      May 28, 2015

         **Interface to Network Security Functions (I2NSF) Problem Statement**
                 **draft-dunbar-i2nsf-problem-statement-05.txt**

Status of this Memo

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

Abstract

   This document describes the motivation and the problem statement for
   Interface to Network Security Functions (I2NSF).

Table of Contents

## 1. Introduction

This document describes the motivation and the problem space for the
Interface to Network Security Functions (I2NSF) effort.

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

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

To meet the demand, more and more service providers are providing
hosted security solutions to deliver cost-effective managed security
services to enterprise customers. The hosted security services are
primarily targeted at enterprises (especially small/medium ones),
but could also be provided to any kind of mass-market customer.

As the result, the Network security functions (NSFs) are provided
and consumed in increasingly diverse environments. Users of NSFs
could consume network security services hosted by one or more
providers, which may be their own enterprise, service providers, or
a combination of both.

This document does not elaborate on specific use case. The reader
should refer to [I2NSF-ACCESS], [I2NSF-DC] and [I2NSF-Mobile] for a
more in-depth discussion on the I2NSF use cases.

## 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 RFC-2119 [RFC2119].

This document makes use of the following terms and acronyms:

DC:         Data Center

Network Security Function (NSF): functions to ensure integrity,
            confidentiality and availability of network
            communications, to detect unwanted activity, and to
            block it or at least mitigate its effects on the
            network.

Hosted security function: Refers to a security function that it is
            hosted by another network.

Flow-based Network Security Function: A function that inspects
            network flows according to a policy intended for

enforcing security properties.  Flow-based security also
means that packets are inspected in the order they are
received, and without modification to the packet due to
the inspection process (MAC rewrites, TTL decrement
action; even NAT would be outside the inspection
process).

## 3. Problem Space

The following sub-sections describe the problems and challenges
facing customers and security service providers (called service
provider, for short) when security functions are no longer
physically hosted by customer's administrative domain.

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

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

3.1. Challenges Facing Security Service Providers

3.1.1. Diverse types of Security Functions

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

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

External Intrusion & Attack Protection:
      e.g., Firewall/ACL; Authentication; IPS; IDS; Endpoint
      Protection; etc;

Security Functions in a DMZ:
        e.g., Firewall/ACL; IDS/IPS, authentication and
        authorization services, NAT, forward proxies, application
        FWs, AAA; etc.

Internal Security Analysis & report:
        e.g., Security Log; Event Correlation; Forensic Analysis;
        etc;

Internal Data and Content Protection:
        e.g., Encryption; Authorization; Public/Private key
        management for internal database, etc.

Given the diversity of security functions, contexts in which they
can be deployed, and constant evolution of these functions,
standardizing all aspects of security functions is challenging,
most probably not feasible, and not necessary. For example, from
an I2NSF perspective, there is no need to standardize on how a
firewall filters are created or applied. What is needed is having
an interface to control and monitor the behavior of NSFs.

3.1.2.  Diverse Interfaces to Control NSFs

To provide effective and competitive solutions and services,
Security Service Providers may need to utilize multiple security
functions from various vendors to enforce the security policies
desired by their customers.

Yet because no widely accepted industry standard security
interfaces exist today, management of NSFs (device and policy
provisioning, monitoring, etc.) tends to be bespoke, essentially
as offered by product vendors. As a result, automation of such
services, if it exists at all, is also bespoke. It is worth noting
that even with the traditional way of deploying security features,
there is still a gap to coordinate among implementations from
distinct vendors. This is mainly the reason why mono-vendor
security functions are enabled in a given network segment.

### 3.1.3. Diverse Interface to monitor the behavior of NSFs

Obviously, enabling a security function (e.g., firewall [I-D.ietf-opsawg-firewalls]) does not mean that a network is protected. As such, it is necessary to have a mechanism to monitor the execution status of NSFs.

### 3.1.4. More Distributed NSFs and vNSFs

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

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

vNSF has higher risk of failure, migrating, and state changes as their hosting VMs being created, moved, or decommissioned.

### 3.1.5. More Demand to Control NSFs Dynamically

In the advent of SDN [SDN-Security], more clients, applications or application controllers need to dynamically update their communication policies that are enforced by NSFs. The Security Service Providers have to dynamically update control requests to NSFs upon receiving the requests from their clients.

### 3.1.6. Demand for multi-tenancy to control and monitor NSFs.

Service providers may require having several operational units to control and monitor the NSFs, especially when NSFs become distributed and virtualized.

### 3.1.7. Lack of Characterization of NSFs and Capability Exchange

To offer effective security services, service providers need to activate various security functions manufactured by multiple

vendors. Even within one product category (e.g., firewall),
security functions provided by different vendors can have
different features and capabilities: filters that can be designed
and activated by a firewall may or may not support IPv6, depending
on the firewall technology, for example.

Service Provider management system (or controller) needs ways to
retrieve the capabilities of service functions by different
vendors so that it could build an effective security solution.

These capabilities can be documented in a static manner or via an
interface for security functions vendors to register to service
provider security management system. This dynamic capability
registration is useful for automation because security functions
may be subject to software and hardware updates. These updates may
have implications on the policies enforced by the NSFs.

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


3.1.8. Lack of mechanism for NSFs to utilize external profiles

Many security functions depend on signature files or profiles to
perform, e.g. IPS/IDS Signatures. Different policies might need
different signatures or profiles. Today, most vendors have their
vendor specific signatures or profiles.  As the industry moves
towards more open environment, sharing profile or black database
can be win-win strategy for all parties involved. There might be
Open Source provided signature/profiles (e.g. by Snort or others)
in the future.

There is a need to have a standard envelop (i.e. the format) to
allow NSFs to use external profiles.

3.2. Challenges Facing Customers

   When customers invoke hosted security services, their security
   policies may be enforced by a collection of security functions
   hosted in different domains. Customers may not have security
   skills. As such, they may not be able to express sufficiently
   precise requirements or security policies. Usually these customers
   express expectations (that can be viewed as loose security
   requirements). Customers may also express guidelines such as which
   critical communications are to be preserved during critical
   events, which hosts are to service even during severe security
   attacks, etc.

3.2.1. NSFs from heterogeneous administrative domains

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

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

3.2.2. Today's Control Requests are Vendors Specific

   Customers may consume NSFs by multiple service providers.
   Customers need to express their security requirements, guidelines,
   and expectations to the service providers, which in turn will be
   translated into security policies and associated configuration
   sets to the set of security functions.  But no standard technical
   characterization and/or APIs exist, even for most common security

services. Most security services are accessible only through
disparate, proprietary interfaces (e.g., portals, APIs), in
whatever format vendors choose to offer.

Without standard interfaces it is complex for customers to update
security policies and integrate with services provided by the
security service providers. This complexity is induced by the
diversity of the configuration models, policy models, supported
management interfaces, etc.

The current practices that rely on the use of scripts that
generates automatically scripts have to be adjusted each time an
implementation from a different vendor is enabled in a provider
side.

Customers may also require means to easily update/modify their
security requirements with immediate effect in the underlying
involved NSFs.

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

Here is an example of how standard interfaces could help achieve
faster implementation time cycles. Let us consider a customer who
would like to dynamically allow an encrypted flow with specific
port, src/dst addresses or protocol type through the firewall/IPS
to enable an encrypted video conferencing call only during the
time of the call. With no commonly accepted interface in place,
the customer would have to learn about the particular provider's
firewall/IPS interface, and send the request in the provider's
required format. If a firewall/IPS interface standard exists, the
customer would be able to send the request, without having to do
much preliminary legwork. Such a standard helps providers too
since they could now offer the same firewall/IPS interface to
represent firewall/IPS services, which may be offered by different
vendors' products. They have now abstracted the firewall/IPS
services. Lastly, it helps the firewall/IPS vendors since they
could now work on common specifications.

### 3.2.3. Difficulty to Monitor the Execution of Desired Policies

How a policy is translated into technology-specific actions is
hidden from the customers. However, customers still need ways to
monitor the delivered security service that is the result of the
execution of their desired security requirements, guidelines and
expectations.

Today, there is no standard way for customers to get security
service assurance (including running "what-if" scenarios to assess
the efficiency of the delivered security service) of their
specified security policies properly enforced by the security
functions in the provider domain.

### 3.3. Difficulty to Validate Policies across Multiple Domains

One key aspect of a hosted security service with security
functions located at different premises is to have a standard
interface to express, monitor and verify security policies that
combine several distributed security functions. This becomes more
crucial when NSFs are instantiated in Virtual Machines because
NSFs can be more distributed and sometimes multiple NSFs are
combined together to perform one task.

Without standard interfaces and security policy data models, the
enforcement of a customer-driven security policy remains
challenging because of the inherent complexity brought by the
combined invocation of several, yet vendor-specific security
functions, but also because of the accompanying complexity of
configuration procedures and operational tasks in a multi-vendor,
heterogeneous environment.

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

3.4. Lack of Standard Interface to Inject Feedback to NSF

   Today, many security functions, such as IPS and Antivirus, depend
   heavily on the associated profiles. They can perform more
   effective protection if they have the up-to-date profiles. As more
   sophisticated threats arise, enterprises, vendors, and service
   providers have to rely on each other to achieve optimal
   protection.  [CA] is one of those initiatives that aim at
   combining efforts conducted by multiple organizations.

   Today there is no standard interface to exchange security profiles
   between organizations.


3.5. Lack of Standard Interface for Capability Negotiation

   There could be situations when the NSFs selected can't perform the
   policies from the Security Controller, due to resource
   constraints. To support the automatic control in the SDN-era, it
   is necessary to have a set of messages for proper negotiation
   between the Security Controller and the NSFs.

**4. Scope of the proposed work**

   The primary goal of I2NSF is to define an information model, a set
   of software interfaces and data models for controlling and
   monitoring aspects of physical and virtual NSFs. Other aspects of
   NSFs, such as device or network provisioning and configuration,
   are out of scope. Controlling and monitoring of NSFs should
   include the ability to specify, query, monitor, and control the
   NSFs by one or more management entities. Since different security
   vendors support different features and functions on their devices,
   I2NSF will focus on flow-based NSFs that provide treatment to
   packets/flows, such as IPS/IDS, Web filtering, flow filtering,
   deep packet inspection, or pattern matching and remediation.

   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. That is,
  I2NSF will standardize a set of interfaces by which control
  and management of NSFs may be invoked, operated, and
  monitored. (I2NSF will not work on any other aspects of NSFs.
  Nor will I2NSF at this stage specify how to derive control
  and monitoring capabilities from higher level security
  policies for the Capability Layer.)
- The I2NSF Service Layer defines how clients' security
  policies may be expressed and monitored. The Service Layer is
  out of scope for this phase of I2NSF's work. However, I2NSF
  will provide a forum for Informational drafts on data models,
  APIs, etc. that demonstrate how service layer policies may be
  translated to Capability Layer functions.

The concrete work at the I2NSF Capability Layer includes
development of
- An information model that defines concepts required for
  standardizing the control and monitoring of NSFs.
- A set of YANG data models, derived from the above information
  model.
- The capability registry (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. The registry enables various
  mechanisms, including policy rules, to be used to match
  monitor and control functions to the needs of an application
  and/or environment.
- The proper secure communication channels to carry the
  controlling and monitoring information between the NSFs and
  their management entity (or entities).


Standard interfaces for monitoring and controlling the behavior of
NSFs are essential building blocks for Security Service Providers
to automate the use of different NSFs from multiple vendors by
their Security management entities. This work will leverage the
existing protocols and data models defined by I2RS, Netconf, and
NETMOD.

I2NSF may be invoked by any (authorized) client-e.g., upstream
applications (controllers), orchestration systems, security
portals, etc.

## [5]. Other Potential Uses of I2NSF

The I2NSF framework allows the clients to view, request, and/or
verify the security functions/policies offered by providers at
different premises. This framework can make it possible for a
cluster of devices requiring the similar security policies to have
consistent policies across multiple sites.

Network service providers can provide "Hosted Security Functions"
services.  Network providers can also act as security function
brokers to facilitate if not optimize the enforcement of customer-
driven security policies. They can expose a service catalog and
standard mechanisms by which enterprises (or applications) can
query, request, or/and verify the needed security functions or
policies.

With the standard interfaces for clients to request the required
security functions and policies, network operators can leverage
their current service to enterprises (e.g. VPN, private IP
services) and access to a vast population of end users to offer a
set of consolidated Security solutions and policies. Network
operators can be instrumental in defining a common interface and
framework as part of an IETF-conducted specification effort.

## [6]. Related Industry Initiatives
### 6.1. Related IETF WGs

IETF NETCONF: I2NSF should consider using the NETCONF protocol
exchange security policy provisioning information between
participating devices/security functions and the computation logic
(a.k.a., a security Policy Decision Point (PDP)) that resides in
the control plane and which makes the decisions to dynamically
allocate resources and enforce customer-driven security policies.

NETMOD ACL Model: [I-D.ietf-netmod-acl-model] describes the very basic attributes for access control. I2NSF will extend the ACL data model to be more comprehensive, for example, extend to multiple actions and policies, and describes various services associated with the security functions under consideration.

In addition, I2NSF has to specify ways to monitor/report of Packet Based Security Functions.

I2RS: the WG currently discusses the specification of an interface between the forwarding and the control planes, to facilitate the dynamic enforcement of traffic forwarding policies based upon IGP/BGP route computation results. I2NSF is looking specifically into expressing security policies in two layers. I2NSF should leverage the protocols and data models developed by I2RS.

I2NSF aims to develop the additional information models and data models for distributed security functions, like the firewall and IPS/IDS. The policy structure specified by [I-D.hares-i2rs-bnp-info-model] can be used by I2NSF to be extended to include recursive actions to other security functions.

The IETF SFC WG specifies service function chaining techniques while treating service functions as a black box; VNFpool is about the reliability and availability of the virtualized network functions. But neither addresses how service functions are invoked, or configured.

Both SFC and VNFpool do not cover in-depth specification (e.g. rules for the requested FW) to invoke security functions. In SFC and VNFpool, a firewall function is a black box that is treated in the same way as a video optimization function. SFC and VNFpool do not cover the negotiation part, e.g. Client needs Rules x/y/z for FW, but the Provider can only offer x/z.

The IETF SACM (Security Assessment and Continuous Monitoring) WG specifies mechanisms to assess endpoint security. The endpoints can be routers, switches, clustered DB, or an installed piece of software. SACM is about "How to encode that policy in a manner where assessment can be automated". For example:

        - a Solaris 10 SPARC or Windows 7 system used in an environment
          that requires adherence to a policy of Mission Critical
          Classified,
        - rules like "The maximum password age must be 30 days" and
          "The minimum password age must be 1 day"

   [I2NSF-GAP] has a more extensive study comparing I2NSF with
   various existing efforts in similar/adjacent areas.

 6.2. Relationship with ETSI NFV ISG

   ETSI's NFV ISG defines the architecture to pool together many
   virtual network functions to be managed and consumed collectively.

   I2NSF is one of the enabling tools for NFV, specifically the VNF
   as a Service (VNFaaS) specified by ETSI NFV Group Specification
   Use Cases [gs_NFV].

   ETSI's NFV ISG effort is actively contributed by service
   providers. It defines a detailed service model for VNFaaS as well
   as requirements that should be taken into account by the I2NSF
   initiative.

 6.3. OpenStack Firewall/Security as a Service

   Open source projects like OpenStack and CloudStack have begun to
   tackle the issues of interfaces to security functions but much
   work remains.

   OpenStack completed the Firewall as a Service project and
   specified the set of APIs for Firewall services [API]

   OpenStack has defined the APIs for managing Security Groups [SG]

   The attributes defined by OpenStack Firewall/Security as a Service
   are at this point are basic. However, they can serve as the basis
   of the information model that the I2NSF IETF initiative aims to
   specify.

6.4.  Security as a Service by Cloud Security Alliance

   https://cloudsecurityalliance.org/research/secaas/#_get-involved

   SaaS by CSA is at the initial stage of defining the scope of work.

## 7.  Manageability Considerations

   Management of NSFs usually include configuration of devices,
   signaling and policy provisioning. I2NSF will only focus on the
   policy provisioning part.

## 8.  Security Considerations

   Having a secure access to control and monitor NSFs is crucial for
   hosted security service. Therefore, proper secure communication
   channels have to be carefully specified for carrying the
   controlling and monitoring information between the NSFs and their
   management entity (or entities).

## 9.  IANA Considerations

   This document requires no IANA actions. RFC Editor: Please remove
   this section before publication.

## 10.  References
### 10.1.  Normative References

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

### 10.2.  Informative References
   [SG] http://docs.openstack.org/admin-guide-
   cloud/content/securitygroup_api_abstractions.html

   [API] http://docs.openstack.org/admin-guide-
   cloud/content/fwaas_api_abstractions.html

   [CA] http://cyberthreatalliance.org/

   [I-D.hares-i2rs-bnp-info-model] Hares, S., Wu, Q., Tantsura, J.,
            and R. White, "An Information Model for Basic Network
            Policy and Filter Rules", draft-hares-i2rs-bnp-info-model-
            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-opsawg-firewalls] Baker, F. and P. Hoffman, "On Firewalls
            in Internet Security", draft-ietf-opsawg-firewalls-01
            (work in progress), October 2012.


   [RFC7297] Boucadair, M., "IP Connectivity Provisioning Profile",
            RFC7297, April 2014.

   [I2NSF-PACKET] E. Lopez, "Packet-based Paradigm for Interfaces to
            NSFs", <draft-lopez-i2nsf-packet-00>, March 2015.

   [I2NSF-ACCESS] A. Pastor, et al, "Access Use Cases for an Open OAM
            Interface to Virtualized Security Services", <draft-
            pastor-i2nsf-access-usecases-00>, Oct 2014.

   [I2NSF-DC] M. Zarny, et al, "I2NSF Data Center Use Cases", <draft-
            zarny-i2nsf-data-center-use-cases-00>, Oct 2014.

   [I2NSF-MOBILE] M. Qi, et al, "Integrated Security with Access
            Network Use Case", <draft-qi-i2nsf-access-network-usecase-
            00>, Oct 2014.

   [SDN-Security] J. Jeong, et al, "Requirement for Security Services
            based on Software-Defined Networking", <draft-jeong-i2nsf-
            sdn-security-services-01>, March 2015.

   [I2NSF-GAP] D. Zhang, et al, "Analysis of Existing Work for I2NSF",
            <draft-zhang-gap-analysis-00>, Feb 2015.

   [gs_NFV] ETSI NFV Group Specification, Network Functions
            Virtualizsation (NFV) Use Cases. ETSI GS NFV 001v1.1.1,
            2013.

   [Gartner-2013] E. Messmer, "Gartner: Cloud-based security as a
            service set to take off", Network World, 31 October 2013

   [NW-2011] J. Burke, "The Pros and Cons of a Cloud-Based Firewall",
            Network World, 11 November 2011

   [Application-SDN] J. Giacomonni, "Application Layer SDN", Layer 123
            ONF Presentation, Singapore, June 2013

11. Acknowledgments

Authors' Addresses
   Linda Dunbar
   Huawei Technologies
   5340 Legacy Drive, Suite 175
   Plano, TX 75024, USA
   Phone: (469) 277 5840
   Email: ldunbar@huawei.com

   Myo Zarny
   Goldman Sachs
   30 Hudson Street
   Jersey City, NJ 07302
   Email: myo.zarny@gs.com

   Christian Jacquenet
   France Telecom
   Rennes 35000
   France
   Email: Christian.jacquenet@orange.com

   Mohamed Boucadair
   France Telecom
   Rennes 35000
   France
   Email: mohamed.boucadair@orange.com

   Shaibal Chakrabarty
   US Ignite
   1776 Massachusetts Ave NW, Suite 601
   Washington, DC 20036
   Phone: (214) 708 6163
   Email: shaibalc@us-ignite.org

 11.1. Appendix: Relationship with Open Source Communities

    One of the goals of the I2NSF initiative is to form a
    collaborative loop from IETF to Industry Open Source Communities.

Open-source initiatives are not to be considered as an alternative
to formal standardization processes. On the contrary, they are
complementary, with the former acting as an enabler and
accelerator of the latter. Open-source provides an ideal mechanism
to quick prototyping and validating contending proposals, and
demonstrating the feasibility of disruptive ideas that could
otherwise not be considered. In this respect, open-source
facilitates the engagement in the standardization process of small
(and typically more dynamic) players such as start-ups and
research groups, which would see better opportunities of being
heard and a clearer rewards to their efforts. An open-source
approach is extremely useful as well for the production of open
reference implementations of the standards at the same (or even
faster) pace they are defined. The availability of such reference
implementations translate into much simpler interoperability and
conformance assessments for both providers and users, and can
become the basis for incremental differentiation of a common
solution, thus allowing a cooperative competition ("coopetition")
model.