

Network Working Group
Internet Draft
Intended status: Informational
Expires: October 2015

L. Dunbar
Huawei
M. Zarny
Goldman Sachs
C. Jacquenet
M. Boucadair
France Telecom
S. Chakrabarty
US Ignite

April 23, 2015

Interface to Network Security Functions (I2NSF) Problem Statement
draft-dunbar-i2nsf-problem-statement-03.txt

Status of this Memo

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

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#). This document may not be modified, and derivative works of it may not be created, except to publish it as an RFC and to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at
<http://www.ietf.org/ietf/1id-abstracts.txt>

The list of Internet-Draft Shadow Directories can be accessed at
<http://www.ietf.org/shadow.html>

This Internet-Draft will expire on October 23, 2015.

Copyright Notice

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

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

Abstract

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

Table of Contents

1. Introduction and Motivation.....	3
2. Requirements Language.....	4
3. Problem Space.....	5
3.1. Challenges facing Customers.....	6
3.1.1. Lack of Standard Interfaces to Express Desired Policies	6
3.1.2. Lack of Interface to Monitor the Execution of Desired Policies.....	8
3.2. Challenges Facing Security Service Providers.....	8
3.2.1. There is no standard Technical Characterization of Security Functions.....	8
3.2.2. Lack of A Standard Catalog of Security Function Capabilities.....	9
3.2.3. Lack of A Common Interface to Enforce Security Policies	10
3.2.4. High-Level Customer's Security Policies.....	10
3.3. Lack of a Clear Interface to validate Policies across Multiple Domains.....	10

3.4. Lack of A Standard Interface to inject feedback to NSF...	11
4. Scope of the proposed work.....	11
4.1. Network Security Functions under Consideration.....	12
4.2. A Two-Layer Approach.....	12
5. Other Potential Uses of I2NSF.....	13
6. Related Industry Initiatives.....	14
6.1. Related IETF WGs.....	14
6.2. Relationship with ETSI NFV ISG.....	15
6.3. OpenStack Firewall/Security as a Service.....	16
6.4. Security as a Service by Cloud Security Alliance.....	16
7. Security Policies Negotiation.....	16
8. Manageability Considerations.....	17
9. Security Considerations.....	17
10. IANA Considerations.....	17
11. References.....	17
11.1. Normative References.....	17
11.2. Informative References.....	17
12. Acknowledgments.....	19
12.1. Appendix: Relationship with Open Source Communities.....	20

1. Introduction and Motivation

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

More and more service providers are providing hosted security solutions to deliver cost-effective managed security services to enterprise customers who face challenges in maintaining a secure infrastructure, complying with regulatory requirements, and controlling costs. 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. The said enterprises often suffer from a lack of security experts who could continuously monitor, acquire new skills and propose immediate mitigations to ever increasing sets of security attacks. Security is a serious concern for the viability of the business to be considered as a part-time job.

However, 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 (Denial of Service (DoS)) attacks. Some enterprises may only need the hosted security services for their remote branch offices where minimal security infrastructures/capabilities exist.

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 leverage existing on-premises security functions and the expertise of service providers to properly select which security features to solicit and to properly configure those functions for a better security protection.

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. Still, 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.

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 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

Hosted Security Function: refers to a security function that is hosted and managed by a third-party (e.g., service providers).

Network Security Function (NSF): denotes a security functions that is located in the network side.

Hosted security function: Refers to a security function that is not located on premises but it is enabled in another administrative domain, typically a service provider's domain, a data center, etc.

Packet based Security Functions: the security functions that perform actions or invoke function calls based content in the packet and/or contextual information such as state, time, events, etc.

3. Problem Space

The following sub-sections describe the problems and challenges facing customers and network security service providers (called service provider, for short) when security functions are no longer physically hosted at customer premises. Security functions can be distributed across networks (or administrative domains): on customer premises or on service provider premises. Security services are then provided by combining several security functions, whether they are located in the customer premises or in the network.

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.

Dynamic means to discover security service functions may not be a valid requirement but means to retrieve the capabilities of on-premises service functions may be required so that a service

provider could propose an efficient security service. These capabilities can be documented in a static manner (e.g., during the contractual agreements) or offer an interface so that a service provider can update the capabilities of security device hosted on the customer's premises. This dynamic capability retrieval is recommended because security functions may be subject to software and hardware updates. These updates may have implications on the policies enforced by the service providers.

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

3.1. 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. The following sub-section elaborates on some customer-specific issues.

Customers may not have security skills. As such, they are not 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.

3.1.1. Lack of Standard Interfaces to Express Desired Policies

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, especially in multi-vendor environments, 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. as implemented by existing implementations. The current practices that rely on the use of scripts that generates automatically scripts should be adjusted each time an implementation from a different vendor is enabled in a network side.

Introducing automated mechanisms to dynamically enforce security polices to accommodate customer's requirement relies upon a formal modeling of security requirements. Note, some customers may express only their general expectations while others may provide more specific security requirements. These requirements are technology-agnostic. Translating these requirements into technology-specific actions is handled by the service provider.

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

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.

3.1.2. Lack of Interface 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.

3.2. Challenges Facing Security Service Providers

Security Service Providers need to utilize multiple instances of security functions from various vendors to enforce the security policies desired by their customers.

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

3.2.1. There is no standard Technical Characterization of Security Functions

Many types of network security functions exist, and they can be deployed in multiple locations in a given network in perhaps different roles. They could be hosted on a common device or on multiple devices. Below are a few examples of security functions and locations/contexts in which they are often deployed.

Security functions can be categorized into meta-domains, such as:

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 the ability for a customer to describe its requirements (intent) of a security policy-possibly by means of APIs.

3.2.2. Lack of A Standard Catalog of Security Function Capabilities

To offer security services, service providers need to activate various security functions on devices 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.

Today, there is no method for vendors to describe the capabilities of their security functions. Without a common technical framework to describe the capabilities of their security functions, various providers could describe security functions in different ways.

3.2.3. Lack of A Common Interface to Enforce Security Policies

Based on customer-specified requirements for global security policy enforcement purposes, service providers need to select a set of security functions (located in the network or elsewhere) to accommodate customer's requirements. However, there are no standardized interfaces to security functions provided by different vendors, making it very difficult to automate the process.

3.2.4. High-Level Customer's Security Policies

Customers may not have security skills. As such, they are not able to express requirements or security policies that are precise enough. 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.

Therefore, service providers have to build system to translate customers' loose and abstract security policies to the exact provisioning data models to the selected security functions.

3.3. Lack of a Clear Interface 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.

The work conducted by ETSI's Network Functions Virtualization (NFV) Industry Specification group (ISG) raises additional management challenges for security policies to be enforced by (distributed) virtual Network Security Functions (vNSF). Virtualization techniques require a standard interface to express, monitor, and manage the security policies that combine several security functions that may be running on different premises, and which may be virtualized or not.

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 A 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.

4. Scope of the proposed work

A primary goal of I2NSF is to define a set of clear interfaces and data models for packet based network security functions (NSFs).

I2NSF will identify how to dynamically design, enforce and manage the security policies that combine several security functions that may be running on different premises.

I2NSF aims to define a template for exposing security requirements. I2NSF also aims at documenting the dynamic security parameter service negotiation procedure to be established between a customer and a service provider (i.e., between a Security Policy

Enforcement Point (SPEP) and a Security Policy Decision Point (SPDP)). The outcomes of such negotiation are meant to feed the computation logic used by the security service provider for dynamic security resource allocation and policy enforcement purposes.

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

4.1. Network Security Functions under Consideration

There are many security functions being deployed and new ones are popping up with business and application demands. In order to have a concrete context for the discussion, I2NSF focuses on the Flow-aware Security Functions that provide treatment to packets/flows, such as IPS/IDS, HTTP filter, and stateless flow filter. (They are different from Application layer security functions, such as email filters, virus treatment, etc). Sample services associated with flow-aware security functions include deep packet inspection, packet/flow/stream filtering, and redirection (remote and local). Sample IPS/IDS functions include flow/stream pattern matching and remediation, respectively.

The reason for starting with flow-based, security-related functions is due to security policies that primarily rely upon security functions that are essentially located in networks.

4.2. A Two-Layer Approach

There are two layers that govern the interaction between network security functions:

- Security Service and Policy Layer
- Functional Layer

The Security Service and Policy Layer is used by customers to express their requirements for a global security policy enforcement and possibly monitor how efficiently the said security

policy is enforced for their specific flows. This layer will leverage existing protocols, such as RESTconf or NETCONF to carry security policy provisioning information that can be expressed by Discretionary Access Control, Mandatory Access Control, Role Based Access Control, Attribute-Based Access Control, Policy-Based Access Control, or combinations of these.

The Functional Layer specifies how customer-driven security policies invoke the security functions that compose a security policy, by means of dynamic configuration procedures. This requires the definition of a vendor-agnostic information model, along with one or more data models, to represent (virtual and physical) security functions that pertain to a security policy. This layer will leverage the existing protocols and data models defined by I2RS, Netconf, and NETMOD WGs.

The security functions offered by hosted security services assume bi-directional information exchange among multiple entities for dynamic policy negotiation and validation purposes. Such exchanges may also yield policy-driven actions, e.g., traffic redirection to higher level security functions, etc. Therefore, the enforcement of security policies requires programmatic interfaces and protocols.

One of the objectives of the proposed work is to standardize the dynamic security service parameter negotiation that typically takes place between the customer and the security service provider to facilitate and contribute to the automation of the overall security service delivery procedure, from service parameter exposure and negotiation to resource allocation and security service fulfillment and assurance.

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 (VNaaS) 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 VNaaS 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. Security Policies Negotiation

The protocol needed for this security function/policies negotiation may be somewhat correlated to the dynamic service parameter negotiation procedure [[RFC7297](#)]. The CPP template documented in [RFC7297](#), even though currently covering only Connectivity (but includes security clauses such as isolation requirements, non-via nodes, etc.), could be extended as a basis for the negotiation

procedure. Likewise, the companion CPNP could be a candidate to proceed with the negotiation procedure.

The "security as a service" would be a typical example of the kind of (CPP-based) negotiation procedures that could take place between a corporate customer and a service provider. However, more security specific parameters have to be considered by this proposed work.

8. Manageability Considerations

TBD.

9. Security Considerations

This document sketches a problem statement for the dynamic interaction with service functions.

10. IANA Considerations

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

11. References

11.1. Normative References

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

11.2. Informative References

[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.

[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
Virtualization (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. Giacomoni, "Application Layer SDN", Layer
123 ONF Presentation, Singapore, June 2013

12. Acknowledgments

Acknowledgments to Andy Malis for his review and contributions.

This document was prepared using 2-Word-v2.0.template.dot.

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

12.1. 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.