

I2NSF
Internet-Draft
Intended status: Informational
Expires: January 29, 2017

S. Hares
J. Strassner
Huawei
D. Lopez
Telefonica I+D
L. Xia
Huawei
July 8, 2016

**Interface to Network Security Functions (I2NSF) Terminology
draft-ietf-i2nsf-terminology-01.txt**

Abstract

This document defines a set of terms that are used for the Interface to Network Security Functions (I2NSF) effort.

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 January 29, 2017.

Copyright Notice

Copyright (c) 2016 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.

Table of Contents

- [1.](#) Introduction [2](#)
- [2.](#) Terminology [3](#)
- [3.](#) IANA Considerations [10](#)
- [4.](#) Security Considerations [10](#)
- [5.](#) Contributors [11](#)
- [6.](#) References [11](#)
 - [6.1.](#) Informative References [11](#)
- Authors' Addresses [12](#)

1. Introduction

This document defines the terminology for the Interface to Network Security Functions (I2NSF) effort. This section provides some background on I2NSF; a detailed problem statement can be found in [[I-D.ietf-i2nsf-problem-and-use-cases](#)]. Motivation and comparison to previous work can be found in [[I-D.ietf-i2nsf-gap-analysis](#)].

Enterprises are now considering using network security functions (NSFs) hosted by service providers due to the growing challenges and complexity in maintaining an up-to-date secure infrastructure that complies with regulatory requirements, while controlling costs. 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. 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 (hosted) 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, but could also be provided to mass-market customers as well. NSFs are provided and consumed in increasingly diverse environments. Users of NSFs may consume network security services hosted by one or more providers, which may be their own enterprise, service providers, or a combination of both.

It is out of scope in this document to define an exhaustive list of terms that are used in the security field; the reader is referred to other applicable documents, such as [[RFC4949](#)].

2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [\[RFC2119\]](#). In this document, these words will appear with that interpretation only when in ALL CAPS. Lower case uses of these words are not to be interpreted as carrying [\[RFC2119\]](#) significance.

3. Terminology

AAA: Authentication, Authorization, and Accounting. See individual definitions.

Abstraction: The definition of the salient characteristics and behavior of an object that distinguish it from all other types of objects. It manages complexity by exposing common properties between objects and processes while hiding detail that is not relevant.

Access Control: Protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy, and is permitted by only authorized entities (users, programs, processes, or other systems) according to that policy [\[RFC4949\]](#).

Accounting: The act of collecting information on resource usage for the purpose of trend analysis, auditing, billing, or cost allocation ([\[RFC2975\]](#) [\[RFC3539\]](#)).

ACL (Access Control List): This is a mechanism that implements access control for a system resource by enumerating the system entities that are permitted to access the resource and stating, either implicitly or explicitly, the access modes granted to each entity [\[RFC4949\]](#). A YANG description is defined in [\[I-D.ietf-netmod-acl-model\]](#).

Action: Defines what is to be done when a set of Conditions are met (See I2NSF Action). (from [\[I-D.ietf-supac-generic-policy-info-model\]](#)).

Assertion: Defined by the ITU in [\[X.1252\]](#) as "a statement made by an entity without accompanying evidence of its validity". In the context of I2NSF, an assertion MAY include metadata about all or part of the assertion (e.g., context of the assertion, or about timestamp indicating the point in time the assertion was created). The validity of an assertion cannot be verified. (from [\[I-D.ietf-sacm-terminology\]](#)).

Authentication: Defined in [[RFC4949](#)] as "the process of verifying a claim that a system entity or system resource has a certain attribute value." (from [[I-D.ietf-sacm-terminology](#)]).

Authorization: Defined in [[RFC4949](#)] as "an approval that is granted to a system entity to access a system resource." (from [[I-D.ietf-sacm-terminology](#)]).

B2B: Business-to-Business.

Bespoke: Something made to fit a particular person, customer, or company.

Bespoke security management: Security management systems that are made to fit a particular customer.

Boolean Clause: A logical statement that evaluates to either TRUE or FALSE. Also called Boolean Expression.

Capability: Defines a set of features that are available from a managed entity (see also I2NSF Capability). Examples of "managed entities" are NSFs and Controllers, where NSF Capabilities and Controller Capabilities define functionality of an NSF and about Controller, respectively. These functions may, but do not have to, be used. All Capabilities are announced through the Registration Interface.

Capability Interface: An interface dedicated to requesting, receiving, editing, and deleting capability information.

Client: See Consumer. [Editor's note: placeholder for gradually replacing Client with Consumer, since Client is too vague and has other connotations (e.g., client-server)].

Client-Facing Interface: See Consumer-Facing Interface.
See also: Interface, NSF-Facing Interface.

Component: An encapsulation of software that communicates using Interfaces. A Component may be implemented by hardware and/or software, and be represented using a set of classes. In general, a Component encapsulates a set of data structures and a set of algorithms that implement the function(s) that it provides.

Consumer: A Consumer is a Role that is assigned to an I2NSF Component that can receive information from another I2NSF Component. See also: Provider, Role.

Consumer-Facing Interface: An Interface dedicated to communication with Consumers of NSF data and Services. This is typically

defined per I2NSF administrative domain. See also: Interface,
NSF-Facing Interface.

Hares, et al.

Expires January 29, 2017

[Page 4]

Condition: A set of attributes, features, and/or values that are to be compared with a set of known attributes, features, and/or values in order to make a decision. A Condition, when used in the context of a Policy Rule, is used to determine whether or not the set of Actions in that Policy Rule can be executed or not. Examples of an I2NSF Condition include matching attributes of a packet or flow, and comparing the internal state of a NSF to a desired state. (from [[I-D.ietf-supra-generic-policy-info-model](#)]).

Constraint: A Constraint is a limitation or restriction. Constraints may be associated with any type of object (e.g., Events, Conditions, and Actions in Policy Rules).

Constraint Programming: A type of programming that uses constraints to define relations between variables in order to find a feasible (and not necessarily optimal) solution.

Context: The Context of an Entity is a collection of measured and/or inferred knowledge that describe the state and the environment in which an Entity exists or has existed. (from <http://www.ietf.org/mail-archive/web/i2nsf/current/msg00762.html>).

Controller: A Controller is a management Component that contains control plane functions to manage and facilitate information sharing, as well as execute security functions. This definition is based on that in [[I-D.ietf-sacm-terminology](#)].

Control Plane: In the context of I2NSF, the Control Plane is an architectural Component that provides common control functions to all I2NSF Components, including some or all of the following: authentication, authorization, accounting, auditing, and Capability discovery and negotiation. The Control Plane orchestrates the operation of the Data Plane according to guidance and/or input from the Management Plane. I2NSF Components with Interfaces to the Control Plane have knowledge of the Capabilities of other I2NSF Components within a particular I2NSF administrative domain. This definition is based on that in [[I-D.ietf-sacm-terminology](#)]. See also: Data Plane, Management Plane.

Customer: A business role of an entity that is involved in the definition and/or consumption of services, and the possible negotiation of a contract to use services from a Provider.

DC: Data Center

Data Model: A representation of concepts of interest to an environment in a form that is dependent on data repository, data definition language, query language, implementation language, and protocol (typically one or more of these). Note the difference between a data **model** and a data **structure**.
[[I-D.ietf-supra-generic-policy-info-model](#)].

Data Plane: In the context of I2NSF, the Data Plane is an architectural Component that provides operational functions to enable an I2NSF Component to provide and consume packets and flows. See also: Control Plane, Management Plane.

Data Structure: A low-level building block that is used in programming to implement an algorithm. A data model typically contains multiple types of data structures; however, a data structure does not contain a data model. Note the difference between a data **model** and a data **structure**.

Event: An important occurrence in time of a change in the system being managed, and/or in the environment of the system being managed. Examples of an I2NSF Event include time and user actions (e.g. logon, logoff, and actions that violate an ACL). An Event, when used in the context of a Policy Rule, is used to determine whether the Condition clause of an imperative Policy Rule can be evaluated or not (from [[I-D.ietf-supra-generic-policy-info-model](#)]).

ECA: Event - Condition - Action (a type of Policy Rule).

Firewall (FW): A function that restricts data communication traffic to and from one of the connected networks (the one said to be 'inside' the firewall), and thus protects that network's system resources against threats from the other network (the one that is said to be 'outside' the firewall) [[RFC4949](#)].
[[I-D.ietf-opsawg-firewalls](#)]

Flow-based NSF: A NSF that inspects network flows according to a set of policies 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.

I2NSF Action: An I2NSF Action is a special type of Action that is used to control and monitor aspects of flow-based Network Security Functions. Examples of I2NSF Actions include providing intrusion detection and/or protection, web and flow filtering, and deep packet inspection for packets and flows. An I2NSF Action, when used in the context of a I2NSF Policy Rule, may be executed when both the Event and the Condition clauses of its owning I2NSF

Policy Rule evaluate to true. The execution of this Action may be influenced by applicable metadata. (from [\[I-D.ietf-supra-generic-policy-info-model\]](#)).

Hares, et al.

Expires January 29, 2017

[Page 6]

I2NSF Agent: A software Component in a device that implements an NSF. It receives provisioning information and requests for operational data (e.g., monitoring data) from an I2NSF Consumer. It is also responsible for enforcing the policies that it receives from an I2NSF Consumer.

I2NSF Capability: A set of features that are available from an NSF Server or an NSF Controller. While both are Capabilities, the former defines functions that are available from an NSF, whereas the latter defines functions that are available from a security Controller or other Management Entity. This definition is based on that in [[I-D.ietf-sacm-terminology](#)].

I2NSF Client: See I2NSF Consumer.

I2NSF Component: A Component that provides one or more I2NSF Services. I2NSF Components are managed and communicate with other I2NSF Components using I2NSF Interfaces.

I2NSF Consumer: A software Component that uses the I2NSF framework to read, write, and/or change provisioning and operational aspects of the NSFs that it attaches to.

I2NSF Consumer Interface: An Interface dedicated to requesting and using I2NSF Services. For example, this Interface could be used to request a set of Flow Security policies from an I2NSF Controller or from one or more individual NSFs. The difference is that the former uses more abstract Condition matching (e.g., based on tenant or customer ID), whereas the latter uses more low-level Condition matching (e.g., based on flow state or fields in a flow or packet). See also: Interface, I2NSF Provider Interface, Client-Facing Interface, NSF-Facing Interface.

I2NSF Management System: I2NSF Consumers operate within the scope of a network management system, which serves as a collection and distribution point for I2NSF security provisioning.

I2NSF Policy: A set of Policy Rules that are used to manage and control the changing or maintaining of the state of an instance of an NSF.

I2NSF Policy Rule: A Policy Rule that is adapted for I2NSF usage. The I2NSF Policy Rule is assumed to be in ECA form (i.e., an imperative structure). Other types of programming paradigms (e.g., declarative and functional) are currently out of scope. An example of an I2NSF Policy Rule is, in pseudo-code:

```
IF <event-clause> is TRUE
  IF <condition-clause> is TRUE
    THEN execute <action-clause>
```

END-IF
END-IF

Hares, et al.

Expires January 29, 2017

[Page 7]

In the above example, the Event, Condition, and Action portions of a Policy Rule are all ****Boolean Clauses****.

I2NSF Provider Interface: An Interface dedicated to providing I2NSF Services. For example, this could provide Anti-Virus, (D)DoS, or IPS Services. See also: Interface, I2NSF Provider Interface, Client-Facing Interface, NSF-Facing Interface.

I2NSF Registry: A registry that contains I2NSF capability information, which can be controlled by the I2NSF Management System. See also: Registry.

I2NSF Service: A set of functions, provided by an I2NSF Consumer, which are used by zero or more I2NSF Producers. Exemplary I2NSF Services include Anti-Virus, Authentication, Authorization, (D)DoS, Firewall, and IPS Services. See also: Interface, I2NSF Provider Interface, Client-Facing Interface, NSF-Facing Interface.

IDS: Intrusion Detection System (see below).

IPS: Intrusion Protection System (see below).

Information Model: A representation of concepts of interest to an environment in a form that is independent of data repository, data definition language, query language, implementation language, and protocol [[I-D.ietf-supra-generic-policy-info-model](#)].

Interface: A set of operations one object knows it can invoke on, and expose to, another object. It is a subset of all operations that a given object implements. The same object may have multiple types of interfaces to serve different purposes. An example of multiple interfaces can be seen by considering the interfaces include a firewall uses; these include:

- * multiple interfaces for data packets to traverse through,
- * an interface for a controller to impose policy, or retrieve the results of execution of a policy rule.

See also: Consumer Interface, I2NSF Interface, Provider Interface

Intrusion Detection System (IDS): A system that detects network intrusions via a variety of filters, monitors, and/or probes. An IDS may be stateful or stateless.

Intrusion Protection System (IPS): A system that protects against network intrusions. An IPS may be stateful or stateless.

Management Plane: In the context of I2NSF, the Management Plane is an architectural Component that provides common functions to define the behavior of I2NSF Components. The primary use of the Management Plane is to transport behavioral commands, and supply OAM data, for making decisions that affect behavior. Examples include modifying the configuration of an I2NSF Component and transporting OAM data. See also: Control Plane, Data Plane.

Metadata: Data that provides information about other data. Examples include IETF network management protocols (e.g. NETCONF, RESTCONF, IPFIX) or IETF routing interfaces (I2RS). The I2NSF security interface may utilize Metadata to describe and/or prescribe characteristics and behavior of the YANG data models.

Middlebox: Any intermediary device performing functions other than the normal, standard functions of an IP router on the datagram path between a source host and destination host [[RFC3234](#)].

Network Security Function (NSF): Software that provides a set of security-related services. Examples include detecting unwanted activity and blocking or mitigating the effect of such unwanted activity in order to fulfil service requirements. The NSF can also help in supporting communication stream integrity and confidentiality.

NSF-Facing Interface: An Interface dedicated to communication with a set of NSFs. This is typically defined per I2NSF administrative domain. See also: Interface, Consumer-Facing Interface.

OAM: Operation, Administrative, and Management.

OCL (Object Constraint Language): A constraint programming language that is used to specify constraints (e.g., in UML) (from <http://www.ietf.org/mail-archive/web/i2nsf/current/msg00762.html>)

Policy Rule: A set of rules that are used to manage and control the changing or maintaining of the state of one or more managed objects. Often this is shortened to Rule or Policy (see I2NSF policy rule). (from [[I-D.ietf-supra-generic-policy-info-model](#)]).

Profile: A structured representation of information that uses a pre-defined set of capabilities of an object, typically in a specific context. Zero or more Capabilities may be changed at runtime. This may be used to simplify how this object interacts with other objects in its environment.

Producer: A Producer is a Role that is assigned to an I2NSF Component that can send information and/or commands to another

I2NSF Component. See also: Consumer, Role.

Hares, et al.

Expires January 29, 2017

[Page 9]

Registry: A logically centralized location containing data of a particular type; it may optionally contain metadata, relationships, and other aspects of the registered data in order to use those data effectively. An I2NSF registry is used to contain capability information that can be controlled by the controller.

Registration Interface: An interface dedicated to requesting, receiving, editing, and deleting information in a Registry.

Role: An abstraction of a Component that models context-specific views and responsibilities of an object as separate Role objects. Role objects can optionally be attached to, and removed from, the object that the Role object describes at runtime. This provides three important benefits. First, it enables different behavior to be supported by the same Component for different contexts. Second, it enables the behavior of a Component to be adjusted dynamically (i.e., at runtime, in response to changes in context) by using one or more Roles to define the behavior desired for each context. Third, it decouples the Roles of a Component from the Applications use that Component.

Service Interface: An Interface dedicated to enabling Policy Rules to be managed. This is also called the I2NSF Consumer Interface.

Service Provider Security Controller: TBD (Editorial: Place holder for a split between controller and security controller definitions.)

Tenant: A group of users that share common access privileges to the same software. An I2NSF tenant may be physical or virtual, and may run on a variety of systems or servers.

Vendor-Facing Interface: An Interface dedicated to registering and vendor-specific NSFs and Capabilities. It is also used to invoke vendor-specific functionality. This is also called the NSF-Facing Interface.

3. IANA Considerations

No IANA considerations exist for this document.

4. Security Considerations

This is a terminology document with no security considerations.

5. Contributors

The following people contributed to creating this document, and are listed in alphabetical order:

Henk Birkholz

6. References

6.1. Informative References

- [I-D.ietf-i2nsf-gap-analysis]
Hares, S., Moskowitz, R., and D. Zhang, "Analysis of Existing work for I2NSF", [draft-ietf-i2nsf-gap-analysis-01](#) (work in progress), April 2016.
- [I-D.ietf-i2nsf-problem-and-use-cases]
Hares, S., Dunbar, L., Lopez, D., Zarny, M., and C. Jacquenet, "I2NSF Problem Statement and Use cases", [draft-ietf-i2nsf-problem-and-use-cases-01](#) (work in progress), July 2016.
- [I-D.ietf-netmod-acl-model]
Bogdanovic, D., Sreenivasa, K., Huang, L., Blair, D., "Network Access Control List (ACL) YANG Data Model", [draft-ietf-netmod-acl-model-08](#) (work in progress), July 2016.
- [I-D.ietf-opsawg-firewalls]
Baker, F. and P. Hoffman, "On Firewalls in Internet Security", [draft-ietf-opsawg-firewalls-01](#) (work in progress), October 2012.
- [I-D.ietf-sacm-terminology]
Birkholz, H., Lu, J., Cam-Wignet, N., "Secure Automation and Continuous Monitoring (SACM) Terminology", [draft-ietf-sacm-terminology-09](#), March 2016
- [I-D.ietf-supra-generic-policy-info-model]
Strassner, J., Halpern, J., and J. Coleman, "Generic Policy Information Model for Simplified Use of Policy Abstractions (SUPA)", [draft-ietf-supra-generic-policy-info-model-00](#) (work in progress), June 2016.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2975] Aboba, B., Arkko, J., and D. Harrington, "Introduction to

Accounting Management", [RFC 2975](#), DOI 10.17487/RFC2975,
October 2000, <<http://www.rfc-editor.org/info/rfc2975>>.

Hares, et al.

Expires January 29, 2017

[Page 11]

- [RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", [RFC 3234](#), DOI 10.17487/RFC3234, February 2002, <<http://www.rfc-editor.org/info/rfc3234>>.
- [RFC3539] Aboba, B. and J. Wood, "Authentication, Authorization and Accounting (AAA) Transport Profile", [RFC 3539](#), DOI 10.17487/RFC3539, June 2003, <<http://www.rfc-editor.org/info/rfc3539>>.
- [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>>.
- [X.1252] ITU-T, "Baseline identity management terms and definitions", Recommendation ITU-T X.1252, April 2010

Authors' Addresses

Susan Hares
Huawei
7453 Hickory Hill
Saline, MI 48176
USA
Phone: +1-734-604-0332
Email: shares@ndzh.com

John Strassner
Huawei Technologies
Santa Clara, CA
USA
Email: john.sc.strassner@huawei.com

Diego R. Lopez
Telefonica I+D
Don Ramon de la Cruz, 82
Madrid 28006
Spain
Email: diego.r.lopez@telefonica.com

Liang Xia (Frank)
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu 210012
China
Email: Frank.Xialiang@huawei.com

