

Interface to Network Security Functions (I2NSF)
Internet-Draft
Intended status: Standards Track
Expires: January 3, 2018

L. Xia
Q. Lin
Huawei
July 2, 2017

Policy Object for Interface to Network Security Functions (I2NSF)
draft-xia-i2nsf-security-policy-object-01

Abstract

This document describes policy object used in the Interface to Network Security Functions (I2NSF) policy rules to provide re-usability and defines essential attributes for each policy object.

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 3, 2018.

Copyright Notice

Copyright (c) 2017 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.

Internet-Draft

Policy Object for I2NSF

July 2017

Table of Contents

1.	Introduction	3
2.	Requirements Language	4
3.	Terminology	4
4.	Policy Object	4
4.1.	Address Object	5
4.1.1.	The addressName Attribute	5
4.1.2.	The addressRange Attribute	5
4.2.	Address Group Object	6
4.2.1.	The addressGroupName Attribute	6
4.2.2.	The addressReference Attribute	6
4.2.3.	The addressRange Attribute	6
4.3.	Service Object	7
4.3.1.	The serviceName Attribute	7
4.3.2.	The serviceList Attribute	7
4.3.2.1.	The serviceProtocol Attribute	7
4.3.2.2.	The serviceProtocolNumber Attribute	8
4.3.2.3.	The serviceICMPType Attribute	8
4.3.2.4.	The serviceICMPCode Attribute	8
4.3.2.5.	The serviceSourcePort Attribute	8
4.3.2.6.	The serviceDestinationPort Attribute	8
4.4.	Service Group Object	8
4.4.1.	The serviceGroupName Attribute	9
4.4.2.	The serviceReference Attribute	9
4.5.	Application Object	9
4.5.1.	The applicationName Attribute	9
4.5.2.	The applicationCategory Attribute	9
4.5.3.	The applicationSubCategory Attribute	9
4.5.4.	The applicationTransmissionModel Attribute	10
4.5.5.	The applicationVulnerability Attribute	10
4.5.6.	The applicationRiskLevel Attribute	10
4.6.	Application Group Object	10
4.6.1.	The applicationGroupName Attribute	10
4.6.2.	The applicationReference Attribute	10
4.7.	Schedule Object	10
4.7.1.	The scheduleName Attribute	11
4.7.2.	The scheduleList Attribute	11
4.7.2.1.	The scheduleType Attribute	11
4.7.2.2.	The scheduleStartTime Attribute	11
4.7.2.3.	The scheduleEndTime Attribute	11
4.7.2.4.	The scheduleWeekDay Attribute	11
4.8.	User Object	11

4.8.1.	The userName Attribute	12
4.8.2.	The userParentGroup Attribute	12
4.8.3.	The userSecurityGroup Attribute	12
4.8.4.	The userDomain Attribute	12
4.8.5.	The userPassword Attribute	13

4.8.6.	The userExpirationTime Attribute	13
4.9.	User Group Object	13
4.9.1.	The userGroupName Attribute	13
4.9.2.	The userGroupParentGroup Attribute	13
4.9.3.	The userGroupDomain Attribute	13
4.9.4.	The userGroupReference Attribute	13
4.10.	Security Group Object	13
4.10.1.	The securityGroupName Attribute	14
4.10.2.	The securityGroupParentGroup Attribute	14
4.10.3.	The securityGroupDomain Attribute	14
4.10.4.	The securityGroupType Attribute	14
4.10.5.	The securityGroupReference Attribute	14
4.10.6.	The securityGroupFilters Attribute	14
5.	Acknowledgements	14
6.	IANA Considerations	14
7.	Security Considerations	14
8.	References	15
8.1.	Normative References	15
8.2.	Informative References	15
Appendix A.	Application Attributes	15
A.1.	Category and Subcategory	15
A.2.	Data Transmission Model	16
A.3.	Vulnerability	17
Appendix B.	Example of Application Scenario for Policy Object	17
B.1.	Security Policy Control for Marketing Departments	20
B.2.	Security Policy Control for R&D Departments	20
B.3.	Security Policy Control for Server Access of Internet Users	21
	Authors' Addresses	21

[1.](#) Introduction

I2NSF policy consists of policy rules that are used to provision NSF instances. The I2NSF policy rule is defined by using "Event-Condition-Action" (ECA) model described in I2NSF framework draft [[I-D.ietf-i2nsf-framework](#)]. In the ECA model, a condition is used to

determine whether or not the predefined actions should be executed. A condition usually consists of several attributes. Information Model of NSFs Capabilities [[I-D.ietf-i2nsf-capability](#)] describes attributes of different condition subclasses. When configuring policy rules by attributes, it is common to see that the same attribute or the same set of several attributes are configured for several times or more. And modifications of the policy rules are also very tedious and time-consuming.

To facilitate the provisioning of NSF instances, this document describes a set of policy objects which are reusable and can be referenced by variable I2NSF policy rules. A policy object consists

of a name attribute that identifies itself and one or several attributes that are typically used together to represent a certain condition. For example, protocol type and port number are usually used together to represent a certain service. Each policy object is predefined and named in order to be used in I2NSF policy rules. By defining policy objects, the creation and maintenance of policy rules are greatly simplified.

- o A policy object can be referenced in different policy rules as required to provide re-usability. And a policy rule can reference several policy objects.
- o The modification of a policy object will be propagated to the I2NSF policy rules that reference this object. No modification should be made to the related policy rules.

In this document, a set of policy objects are described, and for each policy object, several essential attributes are defined.

[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](#)].

[3.](#) Terminology

This document uses the terminology described in Interface to Network Security Functions (I2NSF) Terminology [[I-D.ietf-i2nsf-terminology](#)].

[4.](#) Policy Object

IP addresses, port numbers, protocol types, services, applications, user accounts are commonly used attributes to determine whether a certain condition occurs. In real-world deployment, these attributes are often configured for many times. The definition of policy objects could help to minimize the configuration effort and provide simplicity.

Figure 1 shows the policy objects defined in this document and their relationships.

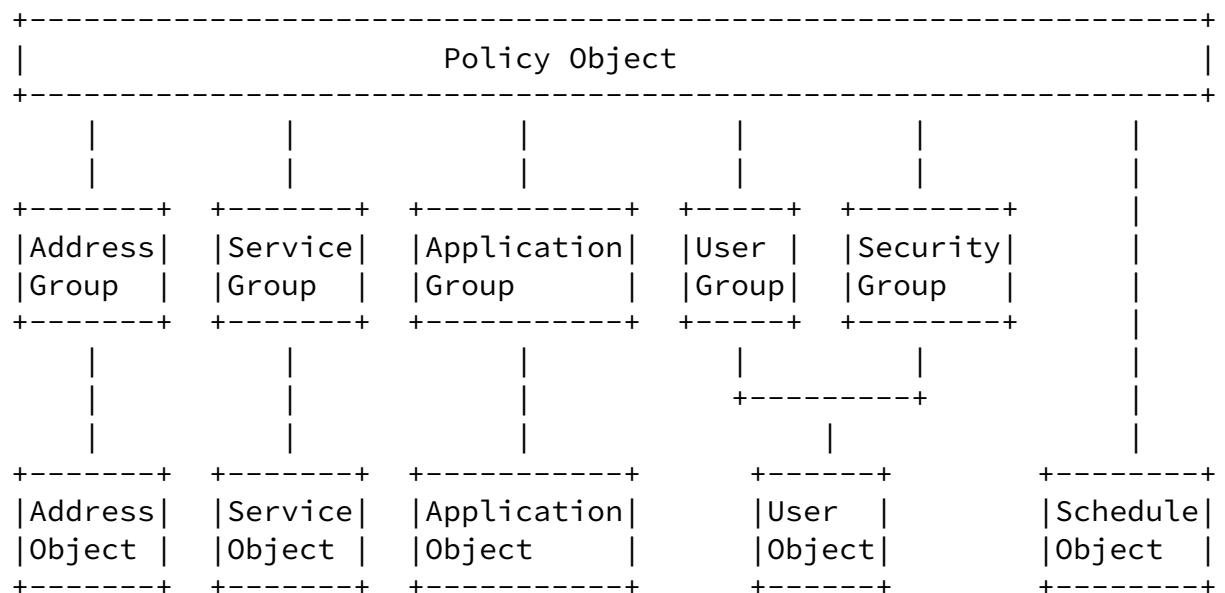


Figure 1: The Policy Objects Overview

[4.1.](#) Address Object

A of IPv4/IPv6 addresses or MAC addresses can be defined as an

address object, which may belongs to an address group object. An address object consists of the following attributes:

[4.1.1.](#) The addressName Attribute

This attribute defines a unique name for the address object.

[4.1.2.](#) The addressRange Attribute

This attribute defines a set of IPv4/IPv6 addresses or MAC addresses, or a range of contiguous IPv4/IPv6 addresses.

An IPv4 address range can be defined by one of the following representations:

- o IPv4 address with wildcard mask, e.g., 10.10.1.2\0.0.0.255.
- o IPv4 address with subnet mask (subnet mask address or length of the subnet mask), e.g., 10.10.1.2/255.255.255.0 or 10.10.1.2/32.
- o Start address and end address of the IPv4 address range, e.g., 10.10.1.2-10.10.1.254.

An IPv6 address range can be defined by one of the following representations:

- o IPv6 address with length of the prefix, e.g., a234::120/120.
- o Start address and end address of the IPv6 address range, e.g., a231::a237-b231::b237.

[4.2.](#) Address Group Object

An address group object is comprised of several address items that require the same policy enforcement. An address item can be an IPv4/IPv6 address, or a MAC address, or a range of contiguous IPv4/IPv6 addresses, or existing address object, or existing address group object. An address group object consists of the following attributes:

[4.2.1.](#) The addressGroupName Attribute

This attribute defines a unique name for the address group object.

[4.2.2.](#) The addressReference Attribute

This attribute refers to the existing address objects or existing address group objects identified by their unique names.

[4.2.3.](#) The addressRange Attribute

This attribute is the same as the addressRange attribute of address object. It can define a set of IPv4/IPv6 addresses or MAC addresses, or a range of contiguous IPv4/IPv6 addresses.

An IPv4 address range can be defined by one of the following representations:

- o IPv4 address with wildcard mask, e.g., 10.10.1.2\0.0.0.255.
- o IPv4 address with subnet mask (subnet mask address or length of the subnet mask), e.g., 10.10.1.2/255.255.255.0 or 10.10.1.2/32.
- o Start address and end address of the IPv4 address range, e.g., 10.10.1.2-10.10.1.254.

An IPv6 address range can be defined by one of the following representations:

- o IPv6 address with length of the prefix, e.g., a234::120/120.
- o Start address and end address of the IPv6 address range, e.g., a231::a237-b231::b237.

[4.3.](#) Service Object

A service object can be a single service based on IP, or ICMP, or UDP, or TCP, or SCTP and it can also contain a set of services. To identify services based on different protocols, different attributes should be specified (see [Section 4.3.2](#) The serviceList Attribute).

- o IP based service is recognized by the value of the protocol field

in IP packet header.

- o ICMP or ICMPv6 based service is recognized by two header fields in the ICMP or ICMPv6 packets: type field and code field.
- o UDP, TCP, or SCTP based service is recognized by port number. The source port number and destination port number are used to identify the sending and receiving service respectively.

A set of well-known services should be predefined by NSFs as service objects to support direct reference. A service object consists of the following attributes:

[4.3.1.](#) The serviceName Attribute

This attribute defines a unique name for the service object.

[4.3.2.](#) The serviceList Attribute

This attribute defined a set of services. Each service can be defined by a subset of the following sub-attributes, according to the protocol on which the service is based.

- o For IP based service, the serviceProtocolNumber attribute should be specified.
- o For ICMP or ICMPv6 based service, the serviceICMPType attribute and serviceICMPCode attribute should be specified.
- o For UDP, TCP, or SCTP based service, the serviceSourcePort attribute and serviceDestinationPort attribute should be specified.

[4.3.2.1.](#) The serviceProtocol Attribute

This attribute defines the protocol type on which the service is based, IP, ICMP, ICMPv6, TCP, UDP, or SCTP.

[4.3.2.2.](#) The serviceProtocolNumber Attribute

This attribute defines the protocol number for IP based service. The protocol number is the value of protocol field in IP packet header which identifies the corresponding upper layer protocol. For example, to define a service object for IPsec Encapsulating Security Payload, this attribute should be set to 50.

[4.3.2.3.](#) The serviceICMPType Attribute

This attribute defines the ICMP/ICMPv6 type number for ICMP/ICMPv6 based service. This attribute shall be used together with serviceICMPCode attribute. For example, to define a service object for IPv4 ping request, this attribute should be set to 8 and serviceICMPCode attribute should be set to 0.

[4.3.2.4.](#) The serviceICMPCode Attribute

This attribute defines the ICMP/ICMPv6 message code for ICMP/ICMPv6 based service. This attribute shall be used together with serviceICMPType attribute. For example, to define a service object for IPv6 ping request, this attribute should be set to 0 and serviceICMPCode should be set to 128.

[4.3.2.5.](#) The serviceSourcePort Attribute

This attribute defines the source port number for service based on TCP, UDP or SCTP. The value could be a single port number which identifies a single service, or a range of port numbers which identify a family of services or several services in consecutive port numbers. For example, to define a service object using port number greater or equal to 1024 and enforce security policy on the traffic that this object sends out, this attribute should be set as a port range, 1024-65535.

[4.3.2.6.](#) The serviceDestinationPort Attribute

This attribute defines the destination port number for service based on TCP, UDP or SCTP. The value could be a single port number or a range of port numbers. For example, to define a service object for HTTP and enforce security policy on the traffic that communicates with this service object, this attribute should be set to 80.

[4.4.](#) Service Group Object

A service group object is a collection of service objects that require the same policy enforcement. It consists of the following attributes:

[4.4.1.](#) The serviceGroupName Attribute

This attribute defines a unique name for the service group object.

[4.4.2.](#) The serviceReference Attribute

This attribute refers to the existing service objects or service group objects identified by their unique names.

[4.5.](#) Application Object

Due to the diversity and large amount of applications, it is not able to identify a certain application based on protocol type and port number. For example, there are many web applications with different risk levels run on ports 80 and 443 using HTTP and HTTPS, such as web gaming application and web chat application. Protocol type and port number could not distinguish applications using the same application protocol. In this document, category, subcategory, data transmission model, vulnerability, and risk level are used to describe an application. A set of well-known application objects should be predefined in NSFs to support direct reference. For a newly created application object, the rules for NSFs to identify this application in the traffic should be configured. In this document, the configuration of these rules is out of scope. An application object consists of the following attributes:

[4.5.1.](#) The applicationName Attribute

This attribute defines a unique name for the application object.

[4.5.2.](#) The applicationCategory Attribute

This attribute defines the category for the application. The value of this attribute is selected from a predefined set of categories, e.g., general category, network category. Values of this attribute are defined in [Appendix A.1](#). Each category is broken down into several subcategories.

[4.5.3.](#) The applicationSubCategory Attribute

This attribute defines the subcategory for the application. The value of this attribute is selected from the predefined subcategories of a category. For example, the entertainment category has seven subcategories, and Facebook application belongs to social networking subcategory. (See [Appendix A.1](#) for details about subcategory and examples of applications belong to each subcategory.)

[4.5.4.](#) The applicationTransmissionModel Attribute

This attribute defines the data transmission model of the application. Four types of data transmission model are defined in this document: client/server, browser-based, network protocol, peer-to-peer. (See [Appendix A.2](#) for more details.)

[4.5.5.](#) The applicationVulnerability Attribute

This attribute describes a set of possible threats for the application. The values of this attribute are selected from a predefined set of vulnerabilities, e.g., exploitable, bandwidth consuming. (See [Appendix A.3](#) for more details.)

[4.5.6.](#) The applicationRiskLevel Attribute

This attribute defines a risk level for the application. The value of this attribute is selected from a predefined number of risk levels, e.g., 5 risk levels. The risk level is determined by the vulnerabilities of this application object.

[4.6.](#) Application Group Object

An application group object is a collection of application objects that will be processed according to the same security policy. It consists of the following attributes:

[4.6.1.](#) The applicationGroupName Attribute

This attribute defines a unique name for the application group object.

[4.6.2.](#) The applicationReference Attribute

This attribute refers to the existing application objects or application group objects identified by their unique names.

[4.7.](#) Schedule Object

A schedule object is a set of time ranges. There are two kinds of time ranges: periodic time range and absolute time range. A periodic time range occurs every week. An absolute time range occurs only once. A schedule object consists of the following attributes:

[4.7.1.](#) The `scheduleName` Attribute

This attribute defines a unique name for the schedule object.

[4.7.2.](#) The `scheduleList` Attribute

This attribute defines a set of time ranges. A time range can be defined by the following sub-attributes.

- o For a periodic time range, the start and end time in a day, and the days in a week that it takes effect, should be specified.
- o For an absolute time range, the start time and date, and the end time and date, should be specified.

[4.7.2.1.](#) The `scheduleType` Attribute

This attribute defines the type of a time range, periodic, absolute.

[4.7.2.2.](#) The `scheduleStartTime` Attribute

For a periodic time range, this attribute defines the start time in a week day, such as 9:00 am. For an absolute time range, this attribute defines the start time and start date, such as 00:00 am 2017-07-03.

[4.7.2.3.](#) The `scheduleEndTime` Attribute

For a periodic time range, this attribute defines the end time in a week day, such as 18:00 pm. For an absolute time range, this attribute defines the end time and end date, such as 23:59 pm 2017-07-03.

[4.7.2.4.](#) The scheduleWeekDay Attribute

This attribute defines the days in a week that the periodic time range takes effect. For example, to define working hours in a week, the scheduleStartTime can be set to 9:00 am, the scheduleEndTime can be set to 18:00 pm, and this attribute should contain fives days, from Monday to Friday.

[4.8.](#) User Object

A user object identifies a person who may access network resources. It is the basis of implementing user-based policy control. The user objects may be created locally on the NSFs, or be imported from third parties, such as authentication servers. User objects that require the same policy enforcement are grouped as user group objects or

security group objects. The user group objects are organized as a hierarchical structure, See Figure 2. A security group object consists of user objects from different user group objects that require the same policy enforcement.

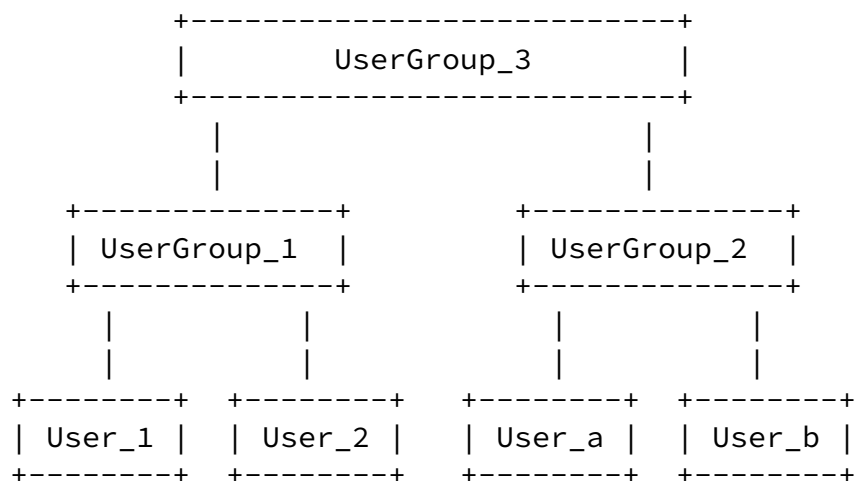


Figure 2: Hierarchical Structure of User Group

A user object consists of the following attributes:

[4.8.1.](#) The userName Attribute

This attribute refers to the user name that used for user authentication.

[4.8.2.](#) The userParentGroup Attribute

This attribute refers to the existing parent user group object to which this user object belongs. The parent user group object is identified by its unique name. A user object can only belong to one user group object.

[4.8.3.](#) The userSecurityGroup Attribute

This attribute refers to the existing security group object to which this user object belongs. The security user group object is identified by its unique name. A user object can belong to several security group objects.

[4.8.4.](#) The userDomain Attribute

This attribute refers to the authentication domain to which this user object belongs.

[4.8.5.](#) The userPassword Attribute

If user is authenticated locally on the NSF, this attribute is mandatory. It defines the password corresponding to the user name.

[4.8.6.](#) The userExpirationTime Attribute

This attribute defines when will this user object expire.

[4.9.](#) User Group Object

A user object group is a collection of user objects that require the same policy enforcement and it usually corresponds to a physical entity such as a department. The user group objects are organized as a hierarchical structure. A user group object may belong to another user group object. The user group objects may be created locally on the NSFs, or be imported from third parties, such as authentication servers. It consists of the following attributes:

[4.9.1.](#) The userGroupName Attribute

This attribute defines a unique name for the user group object.

[4.9.2.](#) The userGroupParentGroup Attribute

This attribute refers to the existing parent user group object to which this user group object belongs. The parent user group object is identified by its unique name. A user group object can only belong to one parent user group object.

[4.9.3.](#) The userGroupDomain Attribute

This attribute refers to the authentication domain to which this user group object belongs.

[4.9.4.](#) The userGroupReference Attribute

This attribute refers to the existing user objects or user group objects which belong to this user group object.

[4.10.](#) Security Group Object

A security group object consists of user objects from different user group objects that require the same policy enforcement. The security group objects may be created locally on the NSFs, or be imported from third parties, such as authentication servers. This attribute consists of the following attributes:

[4.10.1.](#) The securityGroupName Attribute

This attribute defines a unique name for the security group object.

[4.10.2.](#) The securityGroupParentGroup Attribute

This attribute refers to the existing parent security group objects to which this security group object belongs. The parent security group objects are identified by their unique names.

[4.10.3.](#) The securityGroupDomain Attribute

This attribute refers to the authentication domain to which this security group object belongs.

[4.10.4.](#) The securityGroupType Attribute

This attribute defines the type of the security group object. There are two types: static and dynamic. For static security group, the member objects are fixed and added as required. For dynamic security group, the member objects are dynamically generated by setting filtering rules.

[4.10.5.](#) The securityGroupReference Attribute

This attribute defines the member objects for static security group object. It refers to the existing user objects or security group objects which belong to this security group object.

[4.10.6.](#) The securityGroupFilters Attribute

This attribute defines the filtering rules for dynamic security group object.

[5.](#) Acknowledgements

[6.](#) IANA Considerations

This document requires no IANA actions.

[7.](#) Security Considerations

When the policy objects are transmitted, the integrity of these policy objects should be guaranteed. NSFs should verify that the modifications of policy objects come from the authenticated security controller. And NSF should protect the stored policy objects from being tampered.

[8.](#) References

[8.1.](#) Normative References

[I-D.ietf-i2nsf-capability]

Xia, L., Strassner, J., Basile, C., and D. Lopez,
"Information Model of NSFs Capabilities", 2017,
<<https://tools.ietf.org/pdf/draft-xibassnez-i2nsf-capability-01.pdf>>.

[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>>.

[8.2.](#) Informative References

[I-D.ietf-i2nsf-framework]

Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", 2017, <<https://tools.ietf.org/pdf/draft-ietf-i2nsf-framework-05.pdf>>.

[I-D.ietf-i2nsf-terminology]

Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", 2017, <<https://tools.ietf.org/pdf/draft-ietf-i2nsf-terminology-03.pdf>>.

[Appendix A.](#) Application Attributes

An application object is described by five items, category, subcategory, data transmission model, vulnerability and risk level. This appendix illustrates the possible values of applicationCategory attribute, applicationSubCategory attribute, applicationTransmissionModel attribute and applicationVulnerability attribute.

[A.1.](#) Category and Subcategory

This section lists the possible values for applicationCategory attribute and applicationSubCategory attribute.

Category	Subcategory	Example
General	General_TCP	TCP-based applications
	General_UDP	UDP-based applications
	Other	Error_Packets
Network	IP_Protocol	ICMP, IGMP, OSPF
	Encrypted_Tunnel	GRE, L2TP, IKEv2
	Infrastructure	FTP, HTTP, DNS
	Proxy	HTTP_Proxy
	Network_Admin	Syslog
General_Internet	Search_Engine	www.google.com
	Web_Content_Aggregate	FeedReader
	Utility	Google Earth
	Web_Desktop	Zimbra Desktop
	Browser_Plugin	Adobe
	File_Sharing	XDCC
	FileShare_P2P	BT, Thunder
	Network_Storage	DBank
	App_Download	AndroidMarket
	Software_Update	WindowsUpdate
	Web_Browsing	OperaMobile
Entertainment	Social_Networking	Facebook, Twitter
	Instant_Messaging	QQ, MSN
	Media_Sharing	RayV
	Peer_Casting	QQLive
	Web_Video	YouKu, YouTube
	Game	QQGame
	VoIP	Skype
Business_Systems	Electronic_Business	Taobao
	Remote_Access	Radmin
	Database	Oracle
	Finance	DaZhiHui, Fix
	Enterprise_Application	LotusNotes
	Internet_Conferencing	NetMeeting
	Data_Backup	Rsync
	Email	GMail

[A.2.](#) Data Transmission Model

This section lists four types of models for applicationTransmissionModel attribute.

Internet-Draft

Policy Object for I2NSF

July 2017

Model	Description
Client/Server	One or more client applications communicate with a server
Browser-Based	Applications run on web browser
Network Protocol	Applications that is used for system-to-system communication
Peer-to-Peer	Applications directly communicate with each other

[A.3.](#) Vulnerability

This section lists five types of possible risks for applicationVulnerability attribute.

Vulnerability	Description
Exploitable	Has known vulnerabilities
Evasive	Used to evade the original purpose and traverse the firewall, for example, a proxy software
Data Loss	Used for transferring files or uploading texts, may cause information leaks
Used by Malware	Used by malware for propagation, attack, or data theft, or distributed with malware
Bandwidth Consuming	Consume large bandwidths

[Appendix B.](#) Example of Application Scenario for Policy Object

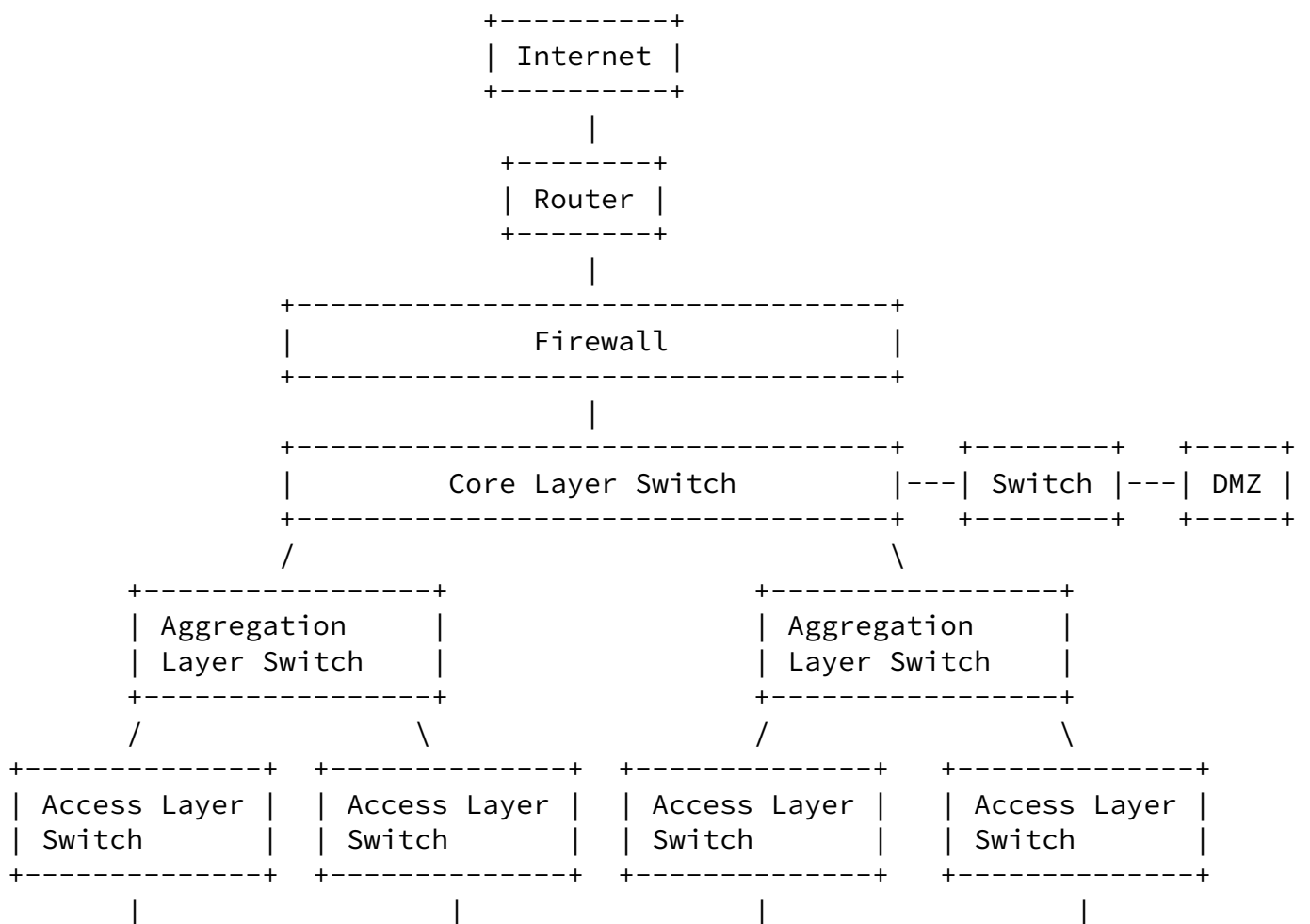
This appendix describes the utilization of policy objects in policy rules for enterprise scenario.

NSFs are key components to protect security in enterprise network. For the typical architecture of an enterprise network, NSFs are deployed on-premise at network perimeter. The inbound and outbound traffic of the enterprise network are processed according to the predefined security policies rules.

Figure 3 demonstrates an example of enterprise network topology. Firewall is a typical NSF that used at the network perimeter to protect enterprise intranet. Assuming that the firewall should be provisioned to provide different network access controls for marketing departments and R&D departments.

- o Marketing departments are allowed to access the Internet website but could not use entertainment applications such as online games, instant messaging software, in work day.
- o R&D departments are not allowed to access the Internet. But managers of R&D departments have Internet access.

For Internet users who want to access the public website of this enterprise, they are only allowed to access the servers deployed in DMZ.



+-----+	+-----+	+-----+	+-----+
Marketing	Marketing	R&D	R&D
Department A	Department B	Department A	Department B
+-----+	+-----+	+-----+	+-----+

Figure 3: A Typical Architecture of Enterprise Network

To set security policy rules for this scenario, the following policy objects should be created.

+-----+	+-----+	+-----+
Policy Object Name	Description	
+-----+	+-----+	+-----+
Marketing_A	User group object for Marketing Department A	
+-----+	+-----+	+-----+
Marketing_B	User group object for Marketing Department B	
+-----+	+-----+	+-----+
R&D_A	User group object for R&D Department A	
+-----+	+-----+	+-----+
R&D_B	User group object for R&D Department B	
+-----+	+-----+	+-----+
R&D_Manager	Security group object for managers of R&D	
	Department A and R&D Department B	
+-----+	+-----+	+-----+
Entertainment_App	Application group object for all recognized	
	entertainment applications	
+-----+	+-----+	+-----+
Server_Address	Address object for servers in DMZ	
+-----+	+-----+	+-----+

Web_Service	Service object for HTTP, HTTPS protocols	
+-----+	+-----+	+-----+
Work_Day	Schedule object for five week days	
+-----+	+-----+	+-----+

[B.1.](#) Security Policy Control for Marketing Departments

For traffic from marketing departments to Internet, the following policy objects can be used as conditions to filter traffic.

+-----+	+-----+
Policy Objects used in Condition	Action
+-----+	+-----+
User Group: Marketing_A, Marketing_B	Deny
Application Group: Entertainment_App	
Schedule: Work_Day	
+-----+	+-----+
User Group: Marketing_A, Marketing_B	Permit
Service: Web_Service	
+-----+	+-----+

[B.2.](#) Security Policy Control for R&D Departments

For traffic from R&D departments to Internet, the following policy objects can be used as conditions to filter traffic.

+-----+	+-----+
Policy Objects used in Condition	Action
+-----+	+-----+
Security Group: R&D_Manager	Permit
+-----+	+-----+
User Group: R&D_A, R&D_B	Deny
+-----+	+-----+

[B.3.](#) Security Policy Control for Server Access of Internet Users

For traffic from Internet to web servers deployed in DMZ, the following policy objects can be used as conditions to filter traffic.

Policy Objects used in Condition	Action
Address: Server_Address	Permit

Authors' Addresses

Liang Xia
Huawei
101 Software Avenue, Yuhuatai District
Nanjing, Jiangsu 210012
China

Email: Frank.xialiang@huawei.com

Qiushi Lin
Huawei
Huawei Industrial Base
Shenzhen, Guangdong 518129
China

Email: linqiushi@huawei.com