Information model for Client-Facing Interface to Security Controller
draft-kumar-i2nsf-client-facing-interface-im-00

Abstract

This document defines information model for the client-facing interface to security controller based on the requirements identified in the [I-D.kumar-i2nsf-client-facing-interface-req]. The information model defines various managed objects and the relationship among these objects needed to build the client interfaces.

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 May 4, 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 .................................................. 3
2. Conventions Used in this Document .......................... 3
3. Information Model for Multi Tenancy .......................... 4
   3.1. Policy-Domain ........................................... 4
   3.2. Policy-Tenant ........................................... 4
   3.3. Policy-Role ............................................. 5
   3.4. Policy-User ............................................. 5
   3.5. Policy-Management-Authentication-method ................. 6
4. Information Model for Policy Endpoint Groups ............... 6
   4.1. Meta Data Source ....................................... 7
   4.2. User-Group ............................................. 7
   4.3. Device-Group ........................................... 8
   4.4. Application-Group ...................................... 8
   4.5. Location-Group ........................................ 9
5. Information Model for Threat Prevention ..................... 9
   5.1. Threat-Feed ............................................. 9
   5.2. Custom-List ............................................ 10
   5.3. Malware-Scan-Group ................................... 10
   5.4. Event-Map-Group ...................................... 11
6. Information Model for Telemetry Data ......................... 11
   6.1. Telemetry-Data ......................................... 11
   6.2. Telemetry-Source ....................................... 12
   6.3. Telemetry-Destination .................................. 12
7. Information Model for Policy Instance ....................... 13
   7.1. Policy-Calendar ........................................ 13
   7.2. Policy-Action .......................................... 14
   7.3. Policy-Rule ........................................... 14
   7.4. Policy-Instance ....................................... 15
8. IANA Considerations ......................................... 15
9. Acknowledgements ............................................ 15
10. Informative References ....................................... 16
Authors’ Addresses ............................................ 16
1. Introduction

The security controller’s client-facing interfaces would be built using a set of objects, with each object capturing a unique set of information from security admin. The objects may have relationship with other objects to express complete requirement. An information model captures the managed objects and relationship among these. The information model proposed in this draft is in accordance with the client interface requirements as defined in [I-D.kumar-i2nsf-client-facing-interface-req].

The [RFC3444] explains the difference between information and data model. This draft use those guidelines to define the information model in this draft. A data model, that represents an implementation of this proposed information model in a specific data representation language, will be defined in a separate draft.

2. Conventions Used in this Document

BSS: Business Support System

CLI: Command Line Interface

CMDB: Configuration Management Database

Controller: Used interchangeably with Service Provider Security Controller or management system throughout this document

CRUD: Create, Retrieve, Update, Delete

FW: Firewall

GUI: Graphical User Interface

IDS: Intrusion Detection System

IPS: Intrusion Protection System

LDAP: Lightweight Directory Access Protocol

NSF: Network Security Function, defined by [I-D.ietf-i2nsf-problem-and-use-cases]

OSS: Operation Support System

RBAC: Role Based Access Control

SIEM: Security Information and Event Management
3. Information Model for Multi Tenancy

The multi-tenancy is an important aspect of any application that enables multiple administrative domains for managing the application resources. An organization may have multiple tenants or departments such as HR, Finance, Legal, with each tenant having a need to manage its own security policies.

There are multiple managed objects that constitute multi-tenancy aspects. This section lists these objects and any relationship among these objects.

3.1. Policy-Domain

This object defines an organization boundary for the purpose of policy management. This may vary based on how security controller is deployed and hosted. For example, if an Enterprise host a security controller in their network; the domain in this case could just be the one that represents that Enterprise. But if a cloud service provider host managed services, then a domain could represent a single customer of the service provider. The multi-tenancy model should be applicable in all such environments.

The domain object SHALL have following information:

Name: Name of the organization or customer representing this domain
Address: Address of the organization or customer
Contact: Contact information of the organization or customer
Date: Date this account was created or last modified
Authentication Method: Authentication method to be used for this domain. It should be reference to a ‘policy-management-auth-method’ object

3.2. Policy-Tenant

This object defines an entity within an organization that wants to manage its own security posture. The entity could be a department or a division that manages its own security policies due to regulatory compliance or organizational structure.
The tenant object SHALL have the following information:

Name:  Name of the department or division within an organization

Date:  Date this account was created or last modified

Domain:  This field identifies the domain to which this tenant belongs. This should be reference to a ‘policy-domain’ object

3.3.  Policy-Role

This object defines a set of permissions assigned to a user in an organization that want to manage its own security posture. It provides a convenient way to assign policy users to a job function or set of permissions within the organization.

The role object SHALL have following information:

Name:  This field identifies the name of the role

Date:  Date this role was created or last modified

Access Profile:  This field identifies the access profile for the role. The profile grants or denies access to policy objects. Multiple access profiles can be concatenated together

3.4.  Policy-User

This object represents a unique identity within an organization. This identity authenticates with the security controller using credentials such as a password or token in order to do policy management. A user may be an individual, system, or application requiring access to security controller.

The user object SHALL have following information:

Name:  Name of the user

Date:  Date this user was created or last modified

Password:  User password for basic authentication

Email:  E-mail address of the user

Scope Type:  This field identifies whether the user has domain-wide or tenant-wide privileges
Scope Reference: This field should be reference to either a ‘policy-domain’ or a ‘policy-tenant’ object

Role: This field should be reference to a ‘policy-role’ object that defines the specific permissions

3.5. Policy-Management-Authentication-method

This object represents authentication schemes supported by security controller.

This object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Authentication Method: This field identifies the authentication methods. It could be a password based, token based, certificate based or single sign-on authentication

Mutual Authentication: This field indicates whether mutual authentication is mandatory or not

Token Server: This field stores the information about server that validates the token submitted as credentials

Certificate Server: This field stores the information about server that validates certificates submitted as credentials

Single Sign-on Server: This field stores the information about server that validates user credentials

4. Information Model for Policy Endpoint Groups

The policy endpoint groups are very important part of building user-construct policy. The security admins could use these objects to represent a logical entity in their business environment, where a security policy is to be applied.

There are multiple managed objects that constitute policy endpoint groups. This section lists these objects and relationship among these objects.
4.1. Meta Data Source

This object represents information source for the meta-data or tag. The meta-data in a group must be mapped to corresponding contents to enforce a security policy.

The meta data source object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Tag Type: This field identifies the Endpoint group type. It can be either a 'User' group or 'App' group or 'Device' group, or 'Location' group based policy

Tag Server Information: This field identifies the information related to the source of the tag such as IP address and UDP/TCP port information

Tag Application Protocol: This field identifies the protocol e.g. LDAP, Active Directory, or CMDB

Tag Server Credentials: This field identifies the credential information needed to access the tag server

4.2. User-Group

This object represents a user group based on either tag or other information.

The user-group object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Group Type: This field identifies whether the user group is based on 'User-tag', 'User-names', or 'IP-addresses'

Meta-data Server: This field should be reference to a 'meta-data-source' object

Group Member: This field is the 'User-tag', or 'User-names', or IP addresses based on the 'Group Type'

Risk Level: This field represents the threat level; valid range may be 0 to 9
4.3. Device-Group

This object represents a device group based on either tag or other information.

The device-group object SHALL have following information:

Name: This field identifies the name of this object
Date: Date this object was created or last modified
Group Type: This field identifies whether the device group is based on 'Device-tag' or 'Device-names', or IP addresses
Meta-data Server: This field should be reference to a 'meta-data-source' object
Group Member: This field is the 'Device-tag, or 'Device-names', or IP addresses based on the 'Group Type'
Risk Level: This field represents the threat level; valid range may be 0 to 9

4.4. Application-Group

This object represents an application group based on either tag or other information.

The application-group object SHALL have following information:

Name: This field identifies the name of this object
Date: Date this object was created or last modified
Group Type: This field identifies whether the device group is based on 'App-tag' or 'App-names', or IP addresses
Meta-data Server: This field should be reference to a 'meta-data-source' object
Group Member: This field is the 'Device-tag, or 'Device-names', or IP addresses and port information based on the 'Group Type'
Risk Level: This field represents the threat level; valid range may be 0 to 9
4.5. Location-Group

This object represents an location group based on either tag or other information.

The location-group object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Group Type: This field identifies whether the location group is based on ‘Location-tag’ or ‘Location-names’, or IP addresses

Meta-data Server: This field should be reference to a ‘meta-data-source’ object

Group Member: This field is the ‘Location-tag, or ‘Location-names’, or IP addresses based on the ‘Group Type’

Risk Level: This field represents the threat level; valid range may be 0 to 9

5. Information Model for Threat Prevention

The threat prevention plays an important part in the overall security posture by reducing the attack surface. This information could come in the form of threat feeds such as Botnet, GeoIP usually from a third party or external service.

There are multiple managed objects that constitute this category. This section lists these objects and relationship among these objects.

5.1. Threat-Feed

This object represents threat feed such as Botnet servers and GeoIP.

The threat-feed object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Feed Type: This field identifies whether a feed type is IP address based or URL based.
Feed Server: This field identifies the information about the feed provider, it may be an external service or local server

Feed Priority: This field represents the feed priority level to resolve conflict if there are multiple feed sources; valid range may be 0 to 9

5.2. Custom-List

This object represents custom list created for the purpose of defining exception to threat feeds. An organization may want to allow certain exception to threat feeds obtained from a third party

The custom-list object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

List Type: This field identifies whether the list type is IP address based or URL based.

List Property: This field identifies the attributes of the list property e.g. Blacklist or Whitelist.

List Content: This field contains the blacklist or whitelist contents such as IP addresses or URL names.

5.3. Malware-Scan-Group

This object represents information needed to detect malware. This information could come from a local server or uploaded periodically from third party.

The malware-scan-group object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Signature Server: This field contains information about the server from where signatures can be downloaded periodically as updates become available

File Types: This field contains list of file types needed to be scanned for the virus
Malware Signatures: This field contains list of malware signatures or hash

5.4. Event-Map-Group

This object represents an event map containing security events and threat levels used for dynamic policy enforcement.

The event-map-group object SHALL have following information:

Name: This field identifies the name of this object
Date: Date this object was created or last modified
Security Events: This contains a list of security events
Threat Map: This contains a list of threat levels

6. Information Model for Telemetry Data

Telemetry provides visibility into the network activities which can be tapped for further security analytics e.g. detecting potential vulnerabilities, malicious activities etc.

6.1. Telemetry-Data

This object contains information collected for telemetry.

The telemetry-data object SHALL have following information:

Name: This field identifies the name of this object
Date: Date this object was created or last modified
Logs: This field identifies whether ‘Logs’ need to be collected
Syslogs: This field identifies whether ‘Syslogs’ need to be collected
SNMP: This field identifies whether ‘SNMP traps’ and ‘SNMP alarms’ need to be collected
sFlow: This field identifies whether ‘sFlow’ data need to be collected
NetFlow: This field identifies whether ‘NetFlow’ data need to be collected
Interface Stats: This field identifies whether 'Interface' data such as packet bytes and counts need to be collected

6.2. Telemetry-Source

This object contains information related to telemetry source. The source would be a NSF element in the network.

The telemetry-source object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified


NSF Access Parameters: This field contains information such as IP address and protocol (UDP or TCP) port number of the NSF providing telemetry data

NSF Access Credentials: This field contains username and password to authenticate with the NSF

Collection Interval: This field contains time in milliseconds between data collections. For example, value of 5000 means data is streamed to collector every 5 seconds. Value of 0 means data streaming is event-based.

Collection Method: This field contains method of collection whether it is PUSH-based or PULL-based

Heartbeat Interval: This field contains time in seconds the source must send telemetry heartbeat

QoS Marking: This field contains DSCP value source MUST mark on its generated telemetry packets

6.3. Telemetry-Destination

This object contains information related to telemetry destination. The destination is usually a collector which is either a part of security controller or external system such as SIEM.

The telemetry-destination object SHALL have following information:
Name: This field identifies the name of this object

Date: Date this object was created or last modified

Collector State: This field contains the state info regarding the collector

Collector Access Parameter: This field contains the information such as IP address and protocol (UDP or TCP) port number for the collector’s destination

Collector Access Credentials: This field contains the username and password for the collector

Data Encoding: This field contains the telemetry data encoding, which could in the form of a schema

Data Transport: This field contains streaming telemetry data protocols: whether it is gRPC, protocol buffer over UDP, etc.

7. Information Model for Policy Instance

In order to enforce a security policy, a policy instance must have complete information such as where and when a policy need to be applied. The policy instantiation is done by combining the managed objects described so far and a few others listed below.

7.1. Policy-Calendar

This object contains information related to scheduling a policy. The policy could be activated based on a time calendar or security event including threat level changes.

The Policy Calendar object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Enforecement Type: This field identifies whether the policy enforcement is ‘ADMIN-ENFORCED’ or ‘TIME-ENFORCED’, or ‘EVENT-ENFORCED’

Time Information: This field contains time calendar such as ‘BEGIN-TIME’ and ‘END-TIME’ for one time enforcement or recurring time calendar for periodic enforcement
Event Map: This field contains security events and threat map when this policy need to be activated

7.2. Policy-Action

This object represents actions that a security admin want to perform based on certain traffic class.

The action object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Primary Action: This field identifies the action when a policy rule is matched by NSF. The action could be one of ‘PERMIT’, ‘DENY’, ‘RATE-LIMIT’, ‘TRAFFIC-CLASS’, ‘AUTHENTICATE-SESSION’, ‘IPS’, ‘APP-FIREWALL’

Secondary Action: The security admin can also specify additional action if a rule is matched. This could be one of ‘LOG’, ‘SYSLOG’, ‘SESSION-LOG’

7.3. Policy-Rule

This object represents rules that a security admin want to define in order to express its business objectives through a security policy.

The policy-rule object SHALL have following information:

Name: This field identifies the name of this object

Date: Date this object was created or last modified

Source: This field identifies the source of the traffic. This could be reference to either ‘Policy Endpoint Group’ or ‘threat-feed’ or ‘custom-list’ if security admin wants to specify the source otherwise the default is to match all traffic

Destination: This field identifies the destination of the traffic. This could be reference to either ‘Policy Endpoint Group’ or ‘threat-feed’ or ‘custom-list’ if security admin wants to specify the destination otherwise the default is to match all traffic

Exception: This field identifies the exception consideration when ‘Source’ and ‘Destination’ are matched for a given communication. This should be reference to ‘Policy Endpoint Group’ object
Action: This field identifies the action taken when 'Source' and 'Destination' are matched for a given communication.

Precedence: This field identifies the precedence assigned to this rule by security admin. This is helpful in conflict resolution when two or more rules match a given traffic class.

7.4. Policy-Instance

This object represents a security policy expressed by security admin that would be taken by security controller and enforced on NSF as per the instructions specified in policy instance.

The policy-instance object SHALL have following information:

Name: This field identifies the name of this object.

Date: Date this object was created or last modified.

Rules: This field contains list of rules. If the rule does not have a user defined precedence, then any conflict must be manually resolved.

Scheduling Type: This field specifies when this policy should be scheduled. The policy could be scheduled based on time calendar or event-map.

Scheduling Information: This field contains either the 'Calendar' or 'Event-map' based on 'Schedule type'.

Owner: This field defines the owner of this policy. Only the owner is authorized to modify the the contents of the policy.

8. IANA Considerations

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

9. Acknowledgements

The authors would like to thank Kunal Modasiya, Prakash T. Sehsadri and Srinivas Nimmagadda from Juniper Networks for helpful discussions.
10. Informative References

[I-D.ietf-i2nsf-problem-and-use-cases]

[I-D.ietf-i2nsf-terminology]

[I-D.kumar-i2nsf-client-facing-interface-req]


Authors’ Addresses

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA  94089
US
Email: rkkumar@juniper.net

Anil Lohiya
Juniper Networks
1133 Innovation Way
Sunnyvale, CA  94089
US
Email: alohiya@juniper.net