I2NSF Working Group
Internet-Draft
Intended status: Informational
Expires: January 17, 2018

R. Kumar
A. Lohiya
Juniper Networks
D. Qi
Bloomberg
N. Bitar
S. Palislamovic
Nokia
L. Xia
Huawei

July 16, 2017

Information model for Client-Facing Interface to Security Controller draft-kumar-i2nsf-client-facing-interface-im-03

Abstract

This document defines information model for Client-Facing interface to Security Controller based on the requirements identified in [I-D.ietf-i2nsf-client-facing-interface-req]. The information model defines various managed objects and relationship among these objects needed to build the interface.

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 5
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. Metadata-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 13
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. Security Considerations 16
9. IANA Considerations 16
10. Acknowledgements 16
11. Informative References 16
Authors' Addresses 17

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 needed to express a Security Policy. An object may have relationship with various other objects to express a complete set of requirement. An information model captures the managed objects and relationship among these objects. The information model proposed in this draft is in accordance with interface requirements as defined in [I-D.ietf-i2nsf-client-facing-interface-req].

The [RFC3444] explains differences between an information and data model. This draft use those guidelines to define information model for Client-Facing interface in this draft. A data model, that represents an implementation of the 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-terminology]

OSS: Operation Support System

RBAC: Role Based Access Control

SIEM: Security Information and Event Management

URL: Universal Resource Locator

vNSF: Refers to NSF being instantiated on Virtual Machines

3. Information Model for Multi Tenancy

Multi-tenancy is an important aspect of any application that enables multiple administrative domains in order to manage application resources An Enterprise organization may have multiple tenants or departments such as HR, Finance, Legal, with each tenant having a need to manage their own Security Policies. In a Service Provider, a tenant could represent a Customer that want 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 a boundary for the purpose of policy management within a Security Controller. This may vary based on how the 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 hosts managed services, then a domain could represent a single customer of that Provider. Multi-tenancy model should be able to work in all such environments.

The Policy-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-Authentication-Method' object

3.2. Policy-Tenant

This object defines an entity within an organization. The entity could be a department or business unit within an Enterprise organization that would like to manages its own Policies due to regulatory compliance or business reasons.

The Policy-Tenant object SHALL have 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 Policies. It provides a convenient way to assign policy users to a job function or set of permissions within the organization.

The Policy-Role object SHALL have following information:

Name: This field identifies 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 permissions to access Endpoint Groups for the purpose of policy management or may restrict certain operations related to policy managements.

3.4. Policy-User

This object represents a unique identity within an organization. The identity authenticates with 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 Policy-User object SHALL have following information:

Name: Name of user

Date: Date this user was created or last modified

Password: User password for basic authentication

Email: E-mail address of user

Scope-Type: This field identifies whether a 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 Policy-Management-Authentication-Method object SHALL have following information:

Name: This field identifies 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 Group is very important part of building User-construct based policies. Security Admin would create and 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 Group. This section lists these objects and relationship among these objects.

4.1. Metadata-Source

This object represents information source for metadata or tag. The metadata in a group must be mapped to its corresponding contents to enforce a Security Policy.

Metadata-Source object SHALL have following information:

Name: This field identifies 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 a User-Group, App-Group, Device-Group or Location-Group

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

Tag-Source-Application: This filed identifies the protocol e.g. LDAP, Active Directory, or CMDB used to communicated with server

Tag-Source-Credentials: This field identifies the credential information needed to access the 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-name or IP-address

Metadata-Server: This field should be reference to a Metadata-Source object

Group-Member: This field is a list of User-tag, User-names or IP addresses based on Group-Type

Risk-Level: This field represents the risk level or importance of the Endpoint to Security Admin for policy purpose; the 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-name or IP address

Metadata-Server: This field should be reference to a Metadata-Source object

Group-Member: This field is a list of Device-tag, Device-name or IP address based on Group-Type

Risk-Level: This field represents the risk level or importance of the Endpoint to Security Admin for policy purpose; the 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 application group is based on App-tag or App-name, or IP address

Metadata-Server: This field should be reference to a Metadata-Source object

Group-Member: This field is a list of Application-tag Application-name or IP address and port information based on Group-Type

Risk-Level: This field represents the risk level or importance of the Endpoint to Security Admin for policy purpose; the 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, Location-name or IP address

Metadata-Server: This field should be reference to a Metadata-Source object

Group-Member: This field is a list of Location-tag, Location-name or IP addresses based on Group-Type

Risk Level: This field represents the risk level or importance of the Endpoint to Security Admin for policy purpose; the 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 and GeoIP feeds 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; the 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 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 a 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 used for purpose for Security Policy definition

Threat-Map: This contains a list of threat levels used for purpose for Security Policy definition

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

Log-Data: This field identifies whether Log data need to be collected

Syslog-Data This field identifies whether Syslog data need to be collected

SNMP-Data: This field identifies whether SNMP traps and alarm data need to be collected

sFlow-Record: This field identifies whether sFlow records need to be collected

NetFlow-Record: This field identifies whether NetFlow record need to be collected

NSF-Stats: This field identifies whether statistics need to be collected from NSF

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

Source-Type: This field contains type of the NSF telemetry source: "NETWORK-NSF", "FIREWALL-NSF", "IDS-NSF", "IPS-NSF", "PROXY-NSF or "OTHER-NSF"

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

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

Collection-Interval: This field contains time in milliseconds between each data collection. For example, a 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-Source: This field contains the information such as IP address and protocol (UDP or TCP) port number for the collector's destination

Collector-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 express a Security Policy, a policy instance must have complete information such as where and when a policy need to be applied. The is done by defining a set of managed objects and relationship among them. A policy may be related segmentation, threat mitigation or telemetry data collection from NSF in the network.

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

Enforecment-Type: This field identifies whether the policy enforcement is "ADMIN-ENFORCED", "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 or threat map in order to determine when a policy need to be activated. This is a reference to Evnet-Map-Group defined earlier

7.2. Policy-Action

This object represents actions that a Security Admin want to perform based on certain traffic class.

The Policy-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 rule is matched by NSF. The action could be one of "PERMIT", "DENY", "REDIRECT", "RATE-LIMIT", "TRAFFIC-CLASS", "AUTHENTICATE-SESSION", "IPS", "APP-FIREWALL", or "COLLECT"

Secondary-Action: Security Admin can also specify additional actions if a rule is matched. This could be one of "LOG", "SYSLOG", or "SESSION-LOG"

7.3. Policy-Rule

This object represents rules that a Security Admin want to define in order to express its business objectives in 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, Threat-Feed or Custom-List as defined earlier. This could be a special object "ALL" that match all traffic. This could also be Telemetry-Source for telemetry collection policy.

Destination: This field identifies the destination of the traffic. This could be reference to either Policy-Endpoint-Group, Threat-Feed or Custom-List as defined earlier. This could be a special object "ALL" that match all traffic. This could also be Telemetry-Destination for telemetry collection policy.

Match-Condition: This field identifies the match criteria used to evaluate whether the specified action need to be taken or not. This could be either a Policy-Endpoint-Group identifying a Application set or a set of traffic rules

Match-Direction: This field identifies if the match criteria is to evaluated for both direction of the traffic or only in one direction with default of allowing in the other direction for stateful match conditions. This is optional and by default rule should apply in both directions

Exception: This field identifies the exception consideration when a rule is evaluated for a given communication. This could be reference to "Policy-Endpoint-Group" object or set of traffic matching criteria

Action: This field identifies the action taken when a rule is matched. There is always a implicit action to drop traffic if no rule is matched for a traffic type

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 mechanism to express a Security Policy by Security Admin using Security Controller Client-Facing interface; the policy would be enforced on a NSF.

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 a 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 reference to Policy-Calendar or Event-Map-Group based on Schedule-Type'

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

8. Security Considerations

Information model provides mechanism to protect Client-Facing interface to Security controller. One of the specified mechanism must be used to protect Enterprise network, data and all resources from external attacks. This model mandates that interface must have proper authentication and authorization with Role Based Access Controls to address multi-tenancy requirement. The draft does not mandate that a particular mechanism be used as different organization may have different needs based on their deployment.

9. IANA Considerations

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

10. Acknowledgements

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

11. Informative References

[I-D.ietf-i2nsf-client-facing-interface-req]
Kumar, R., Lohiya, A., Qi, D., Bitar, N., Palislamovic, S., and L. Xia, "Requirements for Client-Facing Interface to Security Controller", draft-ietf-i2nsf-client-facing-interface-req-02 (work in progress), July 2017.

[I-D.ietf-i2nsf-problem-and-use-cases]
Hares, S., Lopez, D., Zarny, M., Jacquenet, C., Kumar, R., and J. Jeong, "I2NSF Problem Statement and Use cases", draft-ietf-i2nsf-problem-and-use-cases-16 (work in progress), May 2017.

[I-D.ietf-i2nsf-terminology]
Hares, S., Strassner, J., Lopez, D., Xia, L., and H. Birkholz, "Interface to Network Security Functions (I2NSF) Terminology", draft-ietf-i2nsf-terminology-04 (work in progress), July 2017.

[RFC3444] Pras, A. and J. Schoenwaelder, "On the Difference between Information Models and Data Models", RFC 3444, DOI 10.17487/RFC3444, January 2003, <http://www.rfc-editor.org/info/rfc3444>.

Authors' Addresses

Rakesh Kumar
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: rakeshkumarcloud@gmail.com

Anil Lohiya
Juniper Networks
1133 Innovation Way
Sunnyvale, CA 94089
US

Email: alohiya@juniper.net

Dave Qi
Bloomberg
731 Lexington Avenue
New York, NY 10022
US

Email: DQI@bloomberg.net

Nabil Bitar
Nokia
755 Ravendale Drive
Mountain View, CA 94043
US

Email: nabil.bitar@nokia.com

Senad Palislamovic
Nokia
755 Ravendale Drive
Mountain View, CA 94043
US

Email: senad.palislamovic@nokia.com

Liang Xia
Huawei
101 Software Avenue
Nanjing, Jiangsu 210012
China

Email: Frank.Xialiang@huawei.com