Internet Engineering Task Force
INTERNET DRAFT

Authors:
R. Rajan, S. Kamat
IBM
P. Bhattacharya
Cisco

26/February/1999

Networking Policy Condition Information Model
draft-ietf-policy-conditions-00.txt

Status of Memo

Copyright Notice

Abstract

This document defines an information model for networking policy conditions as part of the general information model for representing networking policy defined in draft-ietf-policy-core-schema-02.txt. The information model described in this document is focussed on the structural class networkingPolicyCondition that extends the class policyCondition described draft-ietf-policy-core-schema-02.txt. Five auxiliary

classes are defined to describe conditions that refer to (1) the communicating hosts, (2) the communicating users (3) application data (4) routing information at the device enforcing the policy and (5) layer 2 or data link layer information of the device. This document is based on the QoS and IPSec schema first described in [3] and [4].

1. Introduction

This document extends the Policy Framework Core Information Model Class Hierarchy (PFCIM)[1] which presents a schema for representing networking policies. The schema contains five core classes: policyGroup, policyRule, policyCondtion, policyAction, and policyValidityPeriodCondition. The classes comprising the PFCIM are intended to serve as an extensible class hierarchy (through specialization) for defining policy objects that enable application developers, network administrators, and policy administrators to represent policies of different types. Please refer to [1] for details on the classes and their relationships to one another. Policy conditions are meant to represent criteria that administrators use in enforcing control over behavior of devices or users in a network. This document is NOT concerned with all possible conditions that may arise with respect to computing and communication devices. It is particularly targeted at the needs of controlling resource usage and securing communication between users in an IP network. As mandated by the policy working group, the ability to represent policy requirements of integrated services with RSVP, differentiated services and IPSec are the first targets of this effort.

In keeping with the focus of this effort, we identify 5 conditional categories that are commonly used by administrators in controlling access to network resources and services. These are host, user, application and routing and layer 2 or data link layer information. We extend the class policyCondition to the subclass networkingPolicyCondition, and define 5 auxiliary classes: hostConditionAuxClass, userConditionAuxClass, applicationConditionAuxClass, routeConditionAuxClass and layer2ConditionAuxclass. The auxiliary classes may be attached to networkingPolicyCondition in order to create fully formed conditional statements that are appropriately structured for the purpose at hand.

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, reference [5].

2. Extending policyCondition

2.1. The Class Hierarchy

As defined in [1] the class policyCondition inherits from the class top. We extend the class policyCondition to the subclass netwokingPolicyCondition. The class netwokingPolicyCondition has 5 auxiliary classes: hostConditionAuxClass, userConditionAuxClass,

applicationConditionAuxClass, routeConditionAuxClass and layer2ConditionAuxclass.

```
    top
     |
     |
     |----policyCondition
     |      |
     |     networkingPolicyCondition ~~~~~
     |                                  |(auxiliary classes)
     |-----------------hostConditionAuxClass
     |---------------- userConditionAuxClass
     |-----------------applicationConditionAuxClass
     |-----------------routeConditionAuxClass
     |-----------------layer2ConditionAuxClass
```

2.2. Class Definitions

The class definition of policyCondition, repeated from [1] is as follows:

```
NAME            policyCondition
DESCRIPTION     A class representing a condition to be evaluated
                in conjunction with a policy rule.
DERIVED FROM    top
TYPE            structural
AUXILIARY CLASSES       none
POSSIBLE SUPERIORS      policyRule
OID             <to be assigned>
MUST    cn
        PolicyConditionName
MAY
```

The class policyCondition is specialized to networkingPolicyCondition for extensibility. The class definition is as follows:

```
NAME            networkingPolicyCondition
DESCRIPTION     A class representing a networking condition to be
                evaluated in conjunction with a policy rule.
DERIVED FROM    policyCondition
TYPE            structural
AUXILIARY CLASSES     hostConditionAuxClass
                      userConditionAuxClass
                      applicationConditionAuxClass
                      routeConditionAuxClass
                      layer2ConditionAuxClass
POSSIBLE SUPERIORS    policyRule
```

```
OID                    <to be assigned>
MUST
MAY

   The following auxiliary classes are used to append attributes to the
   class networkingPolicyCondition.

NAME            hostConditionAuxClass
DESCRIPTION     An auxiliary class representing a condition to be
                evaluated in conjunction with a policy rule. The
                condition is based on the communicating end hosts.
DERIVED FROM    top
TYPE            auxiliary
AUXILIARY CLASSES     none
POSSIBLE SUPERIORS    networkingPolicyCondition
OID                   <to be assigned>
MUST
MAY                   sourceIPAddressRange
                      destinationIPAddressRange
                    sourceHostID
                    destinationHostID

NAME                userConditionAuxClass
DESCRIPTION         An auxiliary class representing a condition to be
                    evaluated in conjunction with a policy rule. The
                    condition is based on the communicating users.
DERIVED FROM        top
TYPE                auxiliary
AUXILIARY CLASSES   none
POSSIBLE SUPERIORS  networkingPolicyCondition
OID                 <to be assigned>
MUST
MAY                 senderID
                    receiverID


NAME                applicationConditionAuxClass
DESCRIPTION         An auxiliary class representing a condition to be
                    evaluated in conjunction with a policy rule. The
                    condition is based on the nature of traffic, the
                    transport layer in use and the application.
DERIVED FROM        top
TYPE                auxiliary
AUXILIARY CLASSES   none
POSSIBLE SUPERIORS  networkingPolicyCondition
OID                 <to be assigned>
MUST
MAY                  applicationName
```

```
                     sourcePortRange,
                     destinationPortRange,
                     protocolNumberRange,
                     receivedTOSByteCheck


NAME                 routeConditionAuxClass
DESCRIPTION          An auxiliary class representing a condition to be
                     evaluated in conjunction with a policy rule. The
                     condition is based on the routing of the packet
                     through a device e.g. incoming and outgoing
                     interfaces.
DERIVED FROM         top
TYPE                 auxiliary
AUXILIARY CLASSES    none
POSSIBLE SUPERIORS   networkingPolicyCondition
OID                  <to be assigned>
MUST
MAY                  interface


NAME                layer2ConditionAuxClass
DESCRIPTION         An auxiliary class representing a condition to be
                    evaluated in conjunction with a policy rule. The
                    condition is based on the nature of traffic, the
                    transport layer in use and the application
                    generating data.
DERIVED FROM        top
TYPE                auxiliary
AUXILIARY CLASSES   none
POSSIBLE SUPERIORS  networkingPolicyCondition
OID                 <to be assigned>
MUST
MAY                  sourceMACAddress
                     destinationMACAddress
                     802.1QVLANId
                     SNAPHeaderValue
                     etherTypeValue
                     DSAP
                     SSAP
                     encapsulationType
                     encapsualtionValue
```

2.3. Rationale behind the class structure

There are two design choices that we need to justify in the manner of extending policyCondition. First, the decision to choose particular condition categories and the second the choice of class structures, especially the use of auxiliary classes. The three categories - users, hosts and applications - are very natural to administrative decision making. The need to define policies in terms of a dynamic category such as routing requires some explanation, however. Consider, for instance, a corporation that has its campuses connected by a leased line infrastructure with a backup connection over the internet. When the primary network is down, it is prudent policy to require that inter-campus traffic be encrypted. There are many ways to enforce this, for instance instruct access routers to encrypt traffic based on the destination as well as the outgoing interface. Similar examples may be considered for QoS as well, where a high grade reservation is made over a primary ISP backbone, with a lower grade backup reservation triggered by routing changes. A number of different class hierarchies are feasible even when we have determined the categories we wish to represent, and their attributes. For instance, one choice would have been to associate all the attributes of users, applications, hosts, etc, to the class policyCondition. Why subclass at all? The answer is extensibility. Suppose the same information model is used to represent policy conditions for DHCP. While we would like to have host attributes to express this condition, layer 2 attributes may be totally irrelevant. The subclass networkingPolicyCondition allows us to group all the conditions required for the purpose of expressing networking policy without requiring that all extensions have the same condition attributes. Now the design choice that comes directly from the above is to associate all attributes we want - those of users, hosts, applications, etc - all the class networkingPolicyCondition (as optional attributes, say). Will this not have the same result as the structure presented above? The issue again is extensibility. If one vendor desires to extend the category application, a second only wants to represent users in greater depth, and a third wants to do both, then they dont have to extend networkingPolicyCondition in slightly different ways. Further, the auxiliary classes hostPolicyAuxClass, etc, may be associated with other subclasses of policyCondition, DHCPpolicyCondition for instance, in a selective manner. Finally, the advantage of auxiliary classes is that they allow us to mix and match attributes creating fewer objects, when compared to subclasses.

3. Attributes of HostConditionAuxClass

NAME   sourceIPAddressRange

```
DESC            Source IP addresses to which the policy applies
SYNTAX          IA5String
EQUALITY        caseExactIA5Match
MULTI-VALUED
FORMAT          sourceIPAddressRange may be described in any of the
                following formats.
         1    The string ``1''
                Indicates policy applies to locally generated packets.

         2-<IPv4Address>-<CIDRPrefixLength>
                        Three dash (-) seperated strings. The IP-v4
                        address is in dotted decimal format. The
                        CIDRPrefixLength is the number of unmasked leading
                        bits. A packet matches the condition if the
                        unmasked bits on the packet are identical to the
                        unmasked bits on the condition.

         3-<IPv4Address>-<IPv4Address>
                        IP-v4 addresses in dotted decimal format. The
                        second address must be no smaller than the first.
                        The first denotes the start of the range, and the
                        second denotes the end of the range. A packet
                        matches the condition if its source address is no
                        smaller than the first IP address in the
                        condition, and no larger than the second.

         4-<IPv6Address>-<IPv6Address>
                        IP-v6 addresses in any of the formats supported in
                        RFC 2373. The second address must be no smaller
                        than the first. The first denotes the start of the
                        range, and the second denotes the end of the
                        range. A packet matches the condition if its
                        source address is no smaller than the first
                        address in the condition, and no larger than the
                        second.

DEFAULT                 Defaults to the entire address range, i.e., every
                        packet matches the source address range condition.
EXAMPLES        2-83.23.23.1-24
                A packet with source address 83.23.23.5 matches.
                A packet with source address 83.23.24.1 does not.

                3-83.23.23.0-83.28.28.0
                A packet with source address 83.23.23.5 matches.
                A packet with source address 83.29.24.1 does not.


NAME            destinationIPAddressRange
```

```
DESC            destination IP addresses to which policy applies
SYNTAX          IA5String
EQUALITY        caseExactIA5Match
MULTI-VALUED
FORMAT          Identical to that of sourceIPAddressRange above.
                The value of ``1''indicates a locally destined
                packet.
DEFAULT         Defaults to the entire address range, i.e., every
                packet matches the destination address range
                condition.

NAME            sourceHostID
DESC            Source Host Identifier
SYNTAX          IA5String
EQUALITY        caseExact1A5Match
MULTI-VALUED
FORMAT          Two strings, colon (`:) seperated, the first
                describing the ID type and the second the ID
                value. The following IdTypes and their
                corresponding values are as defined in [2].
                Host-FQDN:<ID>
                X500-DN:<ID>
                X500-GN:<ID>
                Key-Id:<ID>
DEFAULT         Any ID is considered valid.

NAME            destinationHostID
DESC            Destination Host Identifier
SYNTAX         IA5String
EQUALITY        caseExact1A5Match
MULTI-VALUED
DEFAULT         Any ID is considered valid.
FORMAT          Same as SourceHostID.
```

4. Attributes of UserConditionAuxClass

```
NAME            senderID
DESC            Source User Identifier
SYNTAX          IA5String
EQUALITY        caseExact1A5Match
MULTI-VALUED
FORMAT          Two strings colon (:) seperated, the first
                describing the ID type and the second the ID
                value. The following ID Types and their
                corresponding values are as defined in [2].
                User-FQDN:<ID>
                X500-DN:<ID>
```

```
                X500-GN:<ID>
                Key-Id:<ID>
DEFAULT         Any ID is considered valid.

NAME            receiverID
DESC            Destination User Identifier
SYNTAX          IA5String
EQUALITY        caseExact1A5Match
MULTI-VALUED
DEFAULT         Any ID is considered valid.
FORMAT          Same as SourceHostID.
```

5. Attributes of ApplicationConditionAuxClass

```
NAME            sourcePortRange
DESC            Source Ports to which policy applies
SYNTAX          IA5String
EQUALITY        caseExactIA5Match
MULTI-VALUED
FORMAT          String consisting of two colon separated positive
                integers, the second no smaller than the first, or
                one positive integer.
DEFAULT         Defaults to the entire port range 0 to 65535, i.e.
                every packet matches the destination address range
                condition.
EXAMPLE         8000:8080 (ports 8000 to 8080),
                8000 (only port 8000)

NAME            destinationPortRange
DESC            Destination Ports to which policy applies
SYNTAX          IA5String
EQUALITY        caseExactIA5Match
MULTI-VALUED
FORMAT          String consisting of two colon separated positive
                integers, the second no smaller than the first, or
                one positive integer.
DEFAULT        Defaults to the entire port range 0 to 65535, i.e.
                every packet matches the source address range
                condition.

NAME            protocolNumberRange
DESC            Protocol numbers to which policy applies
SYNTAX          INTEGER
EQUALITY        integerMatch
MULTI-VALUED
FORMAT          String consisting of two colon separated positive
                integers, the second no smaller than the first, or
```

```
                 one positive integer.
DEFAULT          Defaults to the entire protocol range 0 to 255,
                 i.e., every packet matches the ip protocol range
                 condition.
EXAMPLE          50:51 (protocol 50 to 51),
                 50 (only protocol 50)

NAME             receivedTOSByteCheck
DESC             A condition attribute used to select traffic based
                 on the contents of the TOS byte of the received
                 packet's IP header
SYNTAX           IA5String
EQUALITY         caseExactIA5Match
MULTI-VALUED
FORMAT           String of the form xxxxxxxx:xxxxxxxx, where each
                 of the x's is either 0 or 1.
SEMANTICS        Each of the substrings is treated as specifying an
                 8-bit field. The left substring is termed Mask and
                 the right substring Match. The TOS byte of the
                 received packet's IP header is ANDed with Mask and
                 the result is compared against Match. The
                 combination of Mask and Match allows definition of
                 TOS byte based conditions where certain bits in
                 the TOS byte may be ignored for the purpose of
                 comparison. Note that <Mask> is superior to
                 <Match> in that TOS bit positions corresponding to
                 a Mask bit of 0 are ignored irrespective of the
                 corresponding <Match> bit.
EXAMPLE          An incoming packet with TOS byte 11001010 matches
                 the condition specified by a value of
                 00111100:00001000 for receivedTOSByte.
```

6. Attributes of RouteConditionAuxClass

```
NAME             interface
DESC             An attribute that limits the scope of the policy
                 to packets on specified  interface(s) and the
                 direction(s) of traffic on these.
SYNTAX           IA5String
EQUALITY         caseExactIA5Match
MULTI-VALUED
FORMAT           Three colon separated strings. The left-most
                 string is a numeral denoting the type of the
                 specification, followed by the incoming and
                 outgoing interface identifiers. Currently defined
                 type/value formats are
```

```
                1-<IPv4Address>-<IPv4Address>
                2-<IPv6Address>-<IPv6Address>
                3-<InterfaceID>-<InterfaceID>

                The IP addresses are in dotted decimal notation.
                The interface IDs are integers unique to the host
                device. The first address string specifies a
                restriction of the rule to traffic inbound on the
                interface, and the rightmost string specifies a
                corresponding restriction of the rule to traffic
                outbound from that interface.  An unspecified
                interface(s) defaults to all interfaces on the
                device that this rule applies to.
DEFAULTS        Defaults to traffic inbound on all interfaces,
                outbound on all interfaces.

EXAMPLE         1-9.3.1.52-9.2.1.54
               (Applies to traffic inbound on 9.3.1.52
                and outbound on 9.3.1.54)

                1-9.3.1.32-
                (Applies to traffic inbound on 9.3.1.52 outbound
                on any interface)

                1- -3
                (Applies to traffic outbound on interface 3 and
                inbound on any interface)
```

7. Attributes of Layer2ConditionAuxClass

```
NAME            sourceMACAddress
DESC            The sourceMACAddress(es) to which the  policy
                applies.
EQUALITY        CaseIgnoreIA5String
SYNTAX          IA5String
MULTI-VALUED
FORMAT          The IEEE Canonical representation of the MAC
                address.
Default         Entire range of values

NAME            destinationMACAddress
DESC            The destination MAC Address(es) to which the
                policy applies.
EQUALITY        CaseIgnoreIA5String
SYNTAX          IA5String
MULTI-VALUED
FORMAT          Same as sourceMACAddress
```

```
Default          Entire range of values

NAME              802.1QVLANID
DESC              The VLAN identified by the value in the 802.1Q
                  VLAN tag.
EQUALITY          IntegerMAtch
SYNTAX            Integer
MULTI-VALUED
FORMAT            The VLAN id in the 802.1Q defined header.
Default           Entire range of values

NAME              SNAPHeaderValue
DESC              The value contained in the SNAP header that
                  identifies the protocol contained in the frame.
EQUALITY          caseIgnoreIA5Match
SYNTAX            IA5String
MULTI-VALUED
FORMAT           A string representing the hexadecimal value that
                 would appear in the SNAP header to identify the
                 protocol.

Default          Entire range of values

NAME             ethertypeValue
DESC            The value contained in the ethertype portion of
                the frame header identifying the protocol
                contained in the frame.
EQUALITY         caseIgnoreIA5Match
SYNTAX           IA5String
MULTI-VALUED
FORMAT           A string representing the hexadecimal value that
                 would appear in the ethertype header to identify
                 the protocol.

NAME             DSAP
DESC             The value contained in the destination SAP of
                 the frame that can be used to identify the frame
                 e.g. a DSAP value of 0x04 identifies SNA frames.
EQUALITY         caseIgnoreIA5Match
SYNTAX           IA5String
MULTI-VALUED
FORMAT           A string representing the hexadecimal value that
                 would appear in the DSAP header to identify the
                 protocol.

NAME             SSAP
DESC             The value contained in the source SAP of the frame
                 that can be used to identify the frame
```

```
                e.g a SSAP value of 0x04 identifies SNA frames.
EQUALITY        caseIgnoreIA5Match
SYNTAX          IA5String
MULTI-VALUED
FORMAT          A string representing the hexadecimal value that
                would appear in the SSAP header to identify the
                protocol.
```

8. Security Considerations

There are two potential security considerations, both of which may be addressed through standards compliant mechanisms. The first is the unauthorized access to read or change policy rules and related objects in the directory repository. The schema in this document SHOULD be used in conjunction with an LDAP access control mechanisms. The second exposure for violation of security lies in the communication between policy decision point and the directory repository. Such communication SHOULD be secured, with both ends mutually authenticated using SSL/TLS or IPSec.

Acknowledgments

We would like to acknowledge Debasish Biswas for his original suggestion to use auxiliary classes in this context. We would like to also thank Jean Christophe Martin, Michael See and Skip Booth for many useful suggestions.

References

[1] J. Strassner and E. Ellesson, Policy Framework Core Information Model", draft-ietf-policy-core-schema-01.txt, February 1999.

[2] D. Piper, ``The Internet IP Security Domain Of Interpretation for ISAKMP'', draft-ietf-ipsec-doi-07

[3] Bhattacharya, P., and R. Adams, W. Dixon, R. Pereira, R. Rajan, "An LDAP Schema for Configuration and Administration of IPSec based Virtual Private Networks (VPNs)", Internet-Draft work in progress, October 1998

[4] Rajan, R., and J. C. Martin, S. Kamat, M. See, R. Chaudhury, D. Verma, G. Powers, R. Yavatkar, "Schema for Differentiated Services and Integrated Services in Networks", Internet-Draft work in progress, October 1998

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

AUTHOR'S ADDRESS

Raju Rajan IBM Research 30 Saw Mill River Road Hawthorne, NY 10532
email: raju@watson.ibm.com

Full Copyright Statement