Internet Engineering Task Force   Editors
INTERNET DRAFT   Partha Bhattacharya, IBM
Rob Adams, Cisco
William Dixon, Microsoft
Roy Pereira, Timestep
Raju Rajan, IBM

October 9 1998

An LDAP Schema for Configuration and Administration of IPSec based Virtual Private Networks (VPNs)

draft-ipsec-vpn-policy-schema-00.txt

Status of Memo

Abstract

This document describes the structure of an LDAP directory schema that enables policy based configuration and administration of IPSec based Virtual private networks within and among Internet domains, intranets, and extranets. The schema extends the base IPSec Policy data model in [9] to include end hosts and security gateways. The schema closely follows and expands on the DEN specification [7].

1. Introduction

IPSec [1], [2], [4] is a fairly large and complex protocol requiring participating hosts to negotiate a number of configuration parameters during protocol operation. These parameters typically have security related implications, so that defaults specified in the IPSec documents may not be acceptable to certain end hosts. In such cases, IPSec negotiations would fail and manual intervention would be required. Furthermore, the defaults may lead to redundancies in situations where the end hosts are also performing security operations at a higher layer (e.g. SSL).

The situation becomes more complex if security gateways have to be traversed for two end hosts to communicate. Depending on the end host application, a gateway may either deny or permit the connection or require an IPSec tunnel from either the end host or another gateway acting as a IPSec proxy for the end host. For successful communication, the gateways have to be properly configured to establish IPSec tunnels with certain end hosts and gateways.

In the light of the above discussion, it is plausible that manual configuration of each IPSec host will become less and less viable as more hosts become IPSec enabled. Directory based policy administration is becoming increasingly popular as a versatile and uniform means of managing network services. LDAP [3] is a widely deployed industry standard for accessing directory information. This document presents an LDAP schema for storing IPSec based policy information in a central directory. The schema describes

- the required IP layer security attributes of a connection; i.e. whether the connection should be blocked, permitted in the clear or secured by IPSec,

- end to end IPSec security association attributes in case the connection needs to be secured by IPSec,

- whether security gateways need to be traversed using IPSec; and if so, then the gateway address and the corresponding IPSec security association attributes,

- nested gateway traversal, etc.

We allow policies to be specified for groups of hosts by either specifying groups or ranges of addresses or wildcarded domains. Policies can also be specified by specific user ids as required by IPSec.

The rest of the document is as follows. Section 2 provides general ideas of representing policy rules through a Policy object, the overview of the LDAP schema and the various object classes and their inter-relationships. The schema described above closely follows the policy class hierarchy described in the DEN document [7]. Sections 3-7 details the various objects and their attributes. Section 8 concludes with some VPN scenarios and examples.

1.1.   Specification of Requirements

The keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT", and "MAY" that appear in this document are to be interpreted as described in [Bradner97].

2. Class Hierarchy

In this section, we describe the various classes related to Policy definition, their inheritance hierarchy and inter-relationships. They are best understood within the Common Information Model [8] of the Directory Management Task Force or the directory structure proposed by the Directory Enabled Networks (DEN) specification [7].

```
                 Top
                |----Policy
                |
                |----PolicyCondition
                |       |
                |       |--IPPolicyCondition
                |
                |-----UserIDCondition
                |
                |----PolicyValidityPeriod
                |
                |----PolicyAction
                |        |----RSVPAction
                |        |----DiffServAction
                |        |----ISAKMPAction
                |        |----IPSecSecurityAction
                |
                |----DiffServResourceGroup
                |----RSVPResourceGroup
                |----ISAKMPProposal
                |----IPSecProposal
                |----IPSecTransform
                |----IPSecPrivateDiffieHellmanGroup
                |
                |----PolicyContainmentAuxClass
```

The schema described here closely follows the policy class hierarchy described in the current DEN document [7]. This document expands on the DEN specification but differs in a few significant details, where it was felt that the specification tended to be unclear or redundant.

A structural LDAP object class called Policy is defined as the container for policy rules. An object of this class ``pieces together'' several policy components relating to differentiated services, RSVP and IPSec based VPNs. Only the IPSec related parameters are described here; the RSVP and differentiated services related parameters are described in a related document [6].

A Policy rule is encoded as

      if <PolicyCondition> then <PolicyAction>

A PolicyCondition class specifies attributes that determine when a policy rule applies. These include validity time related parameters and traffic descriptors such as ranges of IP packet header attributes, MAC addresses etc. The policy validity time is described by reference to a PolicyValidityPeriod object that specifies conditions restricting the validity period of a policy rule.

IPPolicyCondition is a subclass of PolicyCondition and describes the conditions based on IP packet header attributes. The reason for subclassing PolicyCondition is to allow extensibility to other networking protocols through sub-classes. Sometimes an IPSec policy needs to be specified by specifiying host or user ids. This is allowed by a reference to a UserIDCondition class that describes a set of ids such as Host FQDN, User FQDN, X.500 name etc.

A PolicyAction class specifies a collection of attributes that detail permissions or additional behaviors that the policy enforcement entity MUST perform when the corresponding policy condition is satisfied. The PolicyAction class is subclassed into a number of protocol or service specific actions -- DiffServAction, RSVPAction, ISAKMPAction and IPSecSecurityAction. The QoS related classes: DiffServAction, RSVPAction, DiffServResourceGroup and RSVPResourceGroup are defined in a related document [6]. This document focuses on the IPSec related classes ISAKMPAction and IPSecSecurityAction.

The ISAKMPAction class specifies attributes required to perform an ISAKMP/Oakley Phase 1 exchange. These include exchange mode, authentication types, Phase 1 proposals, Phase 1 connection management parameters etc. The proposals are described by references to ISAKMPProposal objects. If Private Diffie Hellman groups are to be used in the proposal then an ISAKMPProposal object must contain references to IPSecPrivateDiffieHellmanGroup objects describing private Diffie Hellman groups.

The IPSecSecurityAction class specifies the security action (e.g. permit/deny/secure) for a traffic stream. If the traffic is to be secured by IPSec, then this class specifies attributes required for ISAKMP Phase 2 (or Quick Mode) exchange. These include proxy ids, Phase 2 proposals, and Phase 2 connection management parameters. The Phase 2 proposals are described by references to IPSecProposal objects. An IPSec Proposal consists of logically AND-ed combinations of AH, ESP and IPCOMP protocols. The transform attributes for each protocol are described by references to corresponding IPSecTransform objects.

The modular object design is done to promote the sharing of objects such as IPSecTransforms, IPSecProposals and ISAKMPProposals.

Finally, given a device identity, it must be possible to find all the policies applicable for that device. The auxiliary class PolicyContainmentAuxClass as defined in the DEN specification [7] is for that purpose. It can be attached to a variety of classes that describe devices. The PolicyContainmentAuxClass itself contains an attribute PoliciesContainedRef describing a list of related policies. Therefore the policies for a given device can be obtained by retrieving all the objects specified by the PoliciesContainedRef attribute in an appropriate class such Device (or any other class to which the PolicyContainmentAuxClass class is attached).

3. The class Policy

The Policy object class is the container class for the policy rules. It contains a number of entries, each entry encodes a policy rule that specifies the resources and services that are allowed (or denied) to a stream of packets. An overview of the class Policy is presented below, followed by the detailed sytax and semantics of various attributes.

```
NAME                Policy
TYPE                Structural
DERIVED FROM        Top
MUST
                    CommonName,
                    PolicyScope,
                    PolicyConditionRef,
                    PolicyActionRef,
                    PolicyVersion
MAY
                    PolicyRulePriority,
                    PolicyKeyword,
```

PolicyType,
PolicyName,
PolicyEnabled

The syntax and semantics of the attributes of the class Policy are as follows:

```
NAME      CommonName
DESC      The common name for objects of this class. Used as relative
          distinguished name to identify object within a branch of
          directory tree.
SYNTAX    IA5String
EQUALITY  caseExactIA5Match
SINGLE-VALUED


NAME       PolicyScope
DESC       Lists the services that are controlled through this policy
EQUALITY  caseExactIA5Match
SYNTAX      IA5String
MULTI-VALUED
FORMAT:    The currently defined values for this attribute are:
           DiffServ
           RSVP
           IPSec
           ISAKMP
SEMANTICS: This attribute is used by the appropriate directory clients
           to fetch only those policy rules that are relevant for their
           functionality. The value  DiffServ' means the policy rule
           specifies DiffServ packet classification and traffic treatment.
           The value `RSVP' means specifes an RSVP policy
           decision point. The value `IPSec' means the policy refers
           to an IPSec action rule. The value `ISAKMP' means the policy
           refers to an ISAKMP action rule. Note that this is a multi-
           valued attribute, and the same rule may regulate multiple
           services for a packet stream.


NAME    PolicyConditionRef
DESC    Absolute Distinguished name of LDAP entry of the objectclass
        PolicyCondition, that identify the packets that the policy rule
        applies to.
EQUALITY distinguishedNameMatch
SYNTAX  DN
SINGLE-VALUED
```

The following reference attributes specify the treatment of packets that match the condition specified in the policy rule. The value of a reference attribute is the distinguished name of an LDAP entry which is an object corresponding to a prespecified class. For instance, if the value of the attribute PolicyActionRef is the distinguished name of an entry in the class RSVPAction, then the policy rule specifies the policy relating to the handling of RSVP signalling messages.

NAME   PolicyActionRef

```
DESC      Absolute Distinguished name(s) of LDAP entry, of the objectclass
          PolicyAction, that specifies permissions and restrictions
          that apply to the packets identified by the policy condition
EQUALITY distinguishedNameMatch
SYNTAX     DN
MULTI-VALUED
SEMANTICS Multiple values are understood as logical AND; that is, all
           the actions must be performed
NAME      PolicyVersion
DESC      The version of the policy schema embodied by this policy.
SYNTAX    IA5String
FORMAT    The current draft specifies version ``1.0''
EQUALITY  caseExactIA5Match
SINGLE-VALUED
NAME      PolicyKeyword
DESC      List of keywords that assist in locating this policy
SYNTAX    IA5String
MULTI-VALUED
DEFAULT  No Keywords


NAME      PolicyType
DESC      Describes the types of a policy
SYNTAX    IA5String
MULTI-VALUED
FORMAT    The following values are allowed:
              ISAKMPPhase1
              ISAKMPPhase2
              IPSecDataLocal
              IPSecDataRemote
              RSVPSignalling
              RSVP-DiffServ
              DiffServ


SEMANTICS ISAKMPPhase1 denotes an ISAKMP Phase 1 policy
           ISAKMPPhase2 denotes an ISAKMP Phase 2 or Quick Mode policy
           IPSecDataLocal denotes a policy for securing locally
              originating data by IPSec. Local means either originating
              from the same host or from an host for which this host
              acts as a proxy
           IPSecDataRemote denotes a policy for securing remotely
              originating data by IPSec. Remote is the opposite of Local
              as defined before.
           RSVPSignalling denotes an RSVP signalling policy
           RSVP-DiffServ denotes a policy for mapping an RSVP traffic
              into a DiffServ pipe
           DiffServ denotes a DiffServ policy
```

```
DEFAULT  Unnamed Type


NAME       PolicyName
DESC       A user friendly name of this policy class
SYNTAX     IA5String
SINGLE-VALUED
DEFAULT    No Name
```

The following attribute defines relationships among multiple related rules within the policy repository.

```
NAME     PolicyRulePriority
DESC     Priority level for this rule. Used to resolve ambiguity in
         condition matching when the ranges specified in the Policy
         conditions overlap. Higher values of this attribute imply
         higher priority of the rule.
EQUALITY integerMatch
SYNTAX    INTEGER
DEFAULT  The default value is 0 (lowest priority)
SINGLE-VALUED
SEMANTICS: Whenever a packet matches two rules of different priority,
           the rule with the higher value of PolicyRulePriority is
           applied.


NAME  PolicyEnabled
DESC  A flag describing whether the policy is currently enabled or
       disabled
SYNTAX  IA5String
EQUALITY caseExactIA5Match
SINGLE-VALUED
FORMAT  The currently defined values for this attribute are:
    Enabled
    Disabled
DEFAULT Enabled
```

3.1. PolicyContainmentAuxClass

Policy rules may need to be grouped together for a number of different purposes -- organizational, security, ease of administration, or ease of retrieval by a policy decision point. We reproduce the PolicyContainmentAuxClass from the DEN specification [7] that serves the useful purpose of grouping policies together. This auxillary class definition is as follows:

```
NAME                    PolicyContainmentAuxClass
```

```
TYPE                     Auxillary
DERIVED FROM            Top
AUXILIARY CLASS         None
POSSIBLE SUPERIORS      Organization, OrganizationalUnit, Group,
```

```
                         GroupOfDevices
MUST                    PoliciesContainedRef
MAY
```

The syntax and semantics of its sole attribute are as follows:

```
NAME     PoliciesContainedRef
DESC     Absolute distinguished names of policies grouped together
         for some (context-dependent) purpose.
SYNTAX   DN
EQUALITY distinguishedNameMatch
MULTI-VALUED
```

4. Policy Conditions

In this section we define the abstract class PolicyCondition, its subclass IPPolicyCondition, and the class UserIDCondition. These classes list the conditions that must be statisfied by a stream of packets in order for the referring rule to apply to that packet stream.

The reason for subclassing PolicyCondition is to allow extensibility to other networking protocols through sub-classes such as ATMPolicyCondition (for instance).

```
NAME                    PolicyCondition
TYPE                    Abstract
DERIVED FROM           Top
AUXILIARY CLASS        None
MUST                    CommonName
MAY                     PolicyConditionName
                         PolicyValidityPeriodRef
```

The detailed syntax and semantics of the attributes is as below:

```
NAME      CommonName
DESC      The common name of the policy condition object. Unique within a
          limited scope and used to identify the object within the
          directory tree.
SYNTAX    IA5String
EQUALITY caseExactIA5Match
SINGLE-VALUED
```

```
NAME      PolicyConditionName
DESC      The user friendly name of this entry.The Name related attributes
          are only for ease of user administration.
EQUALITY caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
```

The next attribute is a reference to PolicyValdityPeriod object that identifies the entry that limits the temporal scope of the policy to specified periods of time.

```
NAME      PolicyValidityPeriodRef
DESC      Absolute distinguished name(s) of an PolicyValidityPeriod
          object that specifies the times that the policy is active.
EQUALITY distinguishedNameMatch
MULTI-VALUED
SYNTAX    DN
DEFAULT  Policy applies at all times
```

4.1. The class IPPolicyCondition

The class PolicyCondition is now specialized to deal with IPv4 packet headers in the class IPPolicyCondition.

```
NAME                    IPPolicyCondition
TYPE                    Structural
DERIVED FROM           PolicyCondition
AUXILIARY CLASSES      none
MAY                     Interface,
                         SourceIPAddressRange,
                         DestinationIPAddressRange,
                         SourcePortRange,
                         DestinationPortRange,
                         IPProtocolNumberRange,
                         ReceivedTOSByteCheck
                         HostUserIDRef
```

The first attribute limits the spatial scope of the policy rule by identifying specific router interfaces where the policy is to be applied.

```
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.
EQUALITY caseExactIA5Match
SYNTAX  IA5String
MULTI-VALUED
FORMAT  Three colon seperated 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:<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.


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
                         outgoing from any interface)
          2::3         (Applies to traffic outbound on interface 3
                         arriving on any interface)
DEFAULTS  Defaults to traffic inbound on all interfaces, outbound on
          all interfaces.


NAME      SourceIPAddressRange
DESC      Source IP addresses to which the policy applies
EQUALITY caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
FORMAT    SourceIPAddressRange is of the following form: three colon (':')
          separated strings denoting a range of IP addresses. The
          following formats are currently defined


          1:<IPv4Address>:<CIDRPrefixLength>
                  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.


          2:<IPv4Address>:<IPv4Address>
                  IP 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.


          3
                  Indicates policy applies to locally generated packets.
EXAMPLE       1:83.23.23.1:24
                    A packet with source address 83.23.23.5 matches.
```

```
          A packet with source address 83.23.24.1 does not.
     2:83.23.23.0:83.28.28.0
          A packet with source address 83.23.23.5 matches.
          A packet with source address 83.29.24.1 does not.
DEFAULT       Defaults to the entire address range, i.e., every packet
```

```
               matches the source address range condition.
NAME        DestinationIPAddressRange
DESC        Destination IP addresses to which policy applies
EQUALITY caseExactIA5Match
SYNTAX      IA5String
SINGLE-VALUED
FORMAT     Identical to that of SourceIPAddressRange above.
           The value of ``3''indicates a locally destined packet.
DEFAULT Defaults to the entire address range, i.e., every packet
         matches the destination address range condition.
NAME       SourcePortRange
DESC       Source Ports to which policy applies
EQUALITY caseExactIA5Match
SYNTAX     IA5String
SINGLE-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
EQUALITY caseExactIA5Match
SYNTAX     IA5String
SINGLE-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       IPProtocolNumberRange
DESC       Protocol numbers to which policy applies
EQUALITY integerMatch
SYNTAX     INTEGER
SINGLE-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
EQUALITY caseExactIA5Match
SYNTAX    IA5String
SINGLE-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.
EXAMPLE  An incoming packet with TOS byte 11001010 matches the
          condition specified by a value of 00111100:00001000 for
          ReceivedTOSByte.
NAME    UserIDConditionRef
DESC    Absolute Distinguished name(s) of LDAP entry or entries, of
        an UserIDCondition object that identify the user or
        host whose packets that the policy rule applies to.
EQUALITY distinguishedNameMatch
SYNTAX  DN
MULTI-VALUED
```

4.2. The Class UserIDCondition

In many scenarios, for instance an end host IPSec, policy needs to be specified for a user or a host ID instead of an IP address. A standard example is a corporate worker connecting from home via an ISP. The policy would be specified by Host FQDN, UserFQDN, X500 DN etc. To accomodate this source and destination ids are required.

```
NAME                   HostUserID
TYPE                   Structural
DERIVED FROM          Top
AUXILIARY CLASS       None
MUST                   CommonName
MAY                    SourceID,
                        DestinationID,
NAME       SourceID
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 valid IdTypes and
their corresponding values are defined in [Piper98]. These
include:
Host-FQDN:<ID>
User-FQDN:<ID>
X500-DN:<ID>
X500-GN:<ID>
Key-Id:<ID>
DEFAULT   Any ID is considered valid.

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

5. The class PolicyValidityPeriod

Objects of this class describe conditions that restrict the validity period of the policy rule. The class definition is as follows:

NAME PolicyValidityPeriod
TYPE Structural
DERIVED FROM Top
AUXILIARY CLASSES NONE
MUST CommonName
MAY   PolicyValidityPeriodName,
PolicyValidityTime,
PolicyValidityMonthMask,
PolicyValidityDayOfWeekMask,
PolicyValidityTimeOfDayRange

The syntax and semantics of various attributes are as given below

NAME   PolicyValidityPeriodName
DESC   The user friendly name of this entry.
EQUALITY caseExactIA5Match
SYNTAX   IA5String
SINGLE-VALUED

```
NAME         PolicyValidityTime
DESC         When this policy is valid
EQUALITY     caseExactIA5Match
```

```
SYNTAX      IA5String
MULTI-VALUED
FORMAT      String(s) of the form yyyymmddhhmmss:yyyymmddhhmmss:<TZ>
SEMANTICS  The first two substrings must be valid times,
            (year-month-date-hour-minute- second) the second larger
            than the first. The last substring is optional and
            describes the time zone.
DEFAULT     If the time zone is omitted then the time is local time at
            the policy decision point. If the attribute itself is absent
            then the policy is always valid.
EXAMPLE     19980121060000:19991231133000:GMT
            (meaning Policy in effect from 6:00 AM on January 21, 1998
            to 1:30 PM on December 31, 1999, Greenwich Mean Time).


NAME      PolicyValidityMonthMask
DESC      Months during which policy is valid
EQUALITY caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
FORMAT    String denoting a mask of 12 0s and 1s.
SEMANTICS 1's corresponding to months in the January to December
           range when the policy is valid.
EXAMPLE  000111111100  (Valid from April until October)
DEFAULT  111111111111, i.e., valid always


NAME      PolicyValidityDayOfWeekMask
DESC      days during which policy is valid
EQUALITY caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
FORMAT    String representing a mask of 7 0s and 1s.
SEMANTICS  1's correspond to days in the Monday to Sunday range
            when the policy is valid.
EXAMPLE  1111100        (Valid weekdays)
DEFAULT  1111111, i.e., valid always


NAME      PolicyValidityTimeOfDayRange
DESC      Time(s) of day during which policy is valid
EQUALITY caseExactIA5Match
SYNTAX    IA5String
MULTI-VALUED
FORMAT    String(s) of the form  hhmmss:hhmmss
SEMANTICS  Substrings on either side of the colon must be be valid
            time of day values. If the second string is not larger
            than the first, then a wrap around midnight is assumed.
EXAMPLE  090000:170000        (Policy valid from 9 AM to 5 PM)
```

DEFAULT 000000:235959

6. The class PolicyAction

While implementing policy within a network device, the PolicyCondition helps identify a substream of packets that are to be granted access to network resources, in a manner that is specified by an instantiation of the class PolicyAction.

The class definition is as follows.

```
NAME  PolicyAction
TYPE  Abstract
DERIVED FROM  Top
AUXILIARY CLASSES  NONE
MUST  CommonName
```

The PolicyAction is subclassed into a number of protocol or service specific actions, each of which is described next.

6.1. The class ISAKMPAction

This class describes the ISAKMP/Oakley action attributes for the traffic flow as described by the linked IPPolicyCondition or AuxIDPolicyCondition object.

```
NAME  ISAKMPAction
TYPE  Structural
DERIVED FROM PolicyAction
AUXILIARY CLASSES    NONE
DESC  Describes ISAKMP/Oakley Phase 1 actions
MUST  CommonName,
       ISAKMPExchangeMode,
       ISAKMPProposalRef
MAY
       ISAKMPActionName,
       LocalHostPublicKeyInfo,
       RemoteHostPublicKeyInfo,
       MinSecurityAssociationLifetimeSec,
       MinSecurityAssociationLifetimeKBytes,
       ISAKMPConnectionLifetimeSec,
       ISAKMPConnectionKBytes,
       SecurityAssociationRefreshThreshold,
       ISAKMPConnectionAutoStartFlag
```

```
NAME  ISAKMPActionName
```

```
DESC  The user friendly name of this entry.
EQUALITY  caseExactIA5Match
SYNTAX  IA5String
SINGLE-VALUED
```

The ISAKMPExchangeMode attribute denotes the ISAKMP/Oakley key exchange mode: main or aggressive.

```
NAME  ISAKMPExchangeMode
DESC  ISAKMP-Oakley key Exchange mode
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT The values can be found in [4]
DEFAULT The value corresponding to Main mode
```

The ISAKMPProposalRef attribute describes a set of ordered ISAKMP proposals. Since LDAP does not support ordered lists, the ISAKMPProposalRef attribute is defined as a composite string in order to be able to capture the relative ordering of the proposals.

```
NAME  ISAKMPProposalRef
DESC  Ordered list of absolute DNs of ISAKMPProposal objects
EQUALITY  caseExactIA5Match
SYNTAX  IA5String
MULTI-VALUED
FORMAT  The format is `pref:ISAKMPProposalDN' where
         - pref is an integer denoting the relative preference of
           the proposal. Lower number has higher preference.
         - ISAKMPProposalDN denotes the distinguishing name (DN) of an
           ISAKMPProposal object
```

The following two attributes describe information about the repository of public keys for the source and the destination. The information consists of the type of the public key repository (e.g. Secure DNS, Certificate Authority, LDAP-Directory, Inline ISAKMP), the host name of the public key repository, and acceptable public key certificate encodings.

These are specified as part of policy so that an IPSec host can perform the proper public key operations during an actula ISAKMP/Oakley exchange.

```
NAME  LocalHostPublicKeyInfo
DESC  Information about local hosts's public key. Required for public
       key based Authentication in ISAKMP
EQUALITY  caseIgnoreMatch
SYNTAX  IA5 String
MULTI-VALUED
```

```
FORMAT: A string of the form  `Type : IPName : X500Name: Encoding', where
         - Type is any one of the following types of Public key CAs:
                SecureDNS,
                CA,
                LDAP-Directory
                Inline-ISAKMP
```

```
         - IPName is the fully qualified domain name of the allowed
           certificate authority. It is required for Types `SecureDNS',
           `CA' and  `LDAP-Directory'
         - X500Name is the X500 DN of the CA (for Types `CA' and
           `LDAP-Directory')
         - Encoding is the acceptable certificate when source is using
           Inline ISAKMP to transfer public keys. The following values
           are allowed:
               X.509
               PKCS
               DNS-SIG`KEY
               SPKI
         Multiple values of the attribute is treated as logical OR.
DEFAULT implementation dependent


NAME  RemoteHostPublicKeyInfo
DESC  Information about remote hosts's public key. Required for public
       key based Authentication in ISAKMP
EQUALITY integerMatch
SYNTAX  INTEGER
MULTI-VALUED
FORMAT same as LocalHostPublicKeyInfo
DEFAULT implementation dependent
```

The following two attributes specify minimum ISAKMP security association lifetimes. A received ISAKMP negotiation request with values smaller than this value are rejected.

```
NAME  MinSecurityAssociationLifetimeKBytes
DESC  Minimum Security Association Lifetime in kiloBytes for use in
       ISAKMP negotiation
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT implementation dependent


NAME  MinSecurityAssociationLifetimeSec
DESC  Minimum Security Association Lifetime  in seconds for use in
       ISAKMP negotiation
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT implementation dependent
```

Often it may be desirable to have a long lived ``ISAKMP connection" between two hosts, implying that the ISAKMP Security Associations must be automatically re-negotiated when the (negotiated) security association lifetime expires. The following two attributes specify these values.

```
NAME  ISAKMPConnectionLifetimeKBytes
DESC  A large Lifetime in kiloBytes during which ISAKMP Security
       Associations are periodically renegotiated once they expire
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT The ISAKMP security associations are re-negotiated forever; that
         is the lifetime is infinity


NAME  ISAKMPConnectionLifetimeSec
DESC  A large Lifetime in seconds during which ISAKMP Security
       Associations are periodically renegotiated once they expire
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT The ISAKMP security associations are renegotiated forever; that
         is the lifetime is infinity
```

The SecurityAssociationRefreshThreshold attribute denotes a fraction of negotiated ISAKMP security association Lifetime at which the ISAKMP security association must be refreshed. For example, if the SecurityAssociationRefreshThreshold is 0.9 and the negotiated ISAKMP security association lifetime is 100MBytes, then a new security association must be negotiated when 90 MBytes has been transferred.

```
NAME  SecurityAssociationRefreshThreshold
DESC  Fraction of negotiated ISAKMP Security Association Lifetime at
       which an ISAKMP security association must be refreshed
EQUALITY  caseIgnoreMatch
SYNTAX  IA5String
SINGLE-VALUED
FORMAT a:b where a and b are integers
SEMANTICS a:b means a/b
DEFAULT implementation dependent
```

The ISAKMPConnectionAutoStart flag denotes whether the ISAKMP association must be negotiated at system initialization.

```
NAME  ISAKMPConnectionAutoStartFlag
DESC  Flag denoting whether the ISAKMP security association must be
       automatically negotiated at system initialization
EQUALITY integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT 1 (YES), 0 (NO)
```

DEFAULT 0

6.2. The class IPSecSecurityAction

This class describes the IPSec Security action and related attributes for a traffic flow.

```
NAME  IPSecSecurityAction
TYPE  Structural
DERIVED FROM PolicyAction
AUXILIARY CLASSES NONE
DESC Describes ipsec (Phase 2) security rules
MUST
       CommonName
       SecurityAction
MAY
       IPSecSecurityActionName,
       LocalIPSecTunnelEndPoint,
       RemoteIPSecTunnelEndPoint,
       LocalProxiedAddressRange,
       RemoteProxiedAddressRange,
       LocalProxiedPort,
       RemoteProxiedPort,
       ProxiedIPProtocol,
       ProxiedHostScope,
       IPSecProposalRef,
       ISAKMPActionRef,
       MinSecurityAssociationLifetimeSec,
       MinSecurityAssociationLifetimeKBytes,
       IPSecConnectionLifetimeSec,
       IPSecConnectionLifetimeKBytes,
       SecurityAssociationRefreshThreshold,
       IPSecAutoStartFlag
```

The IPSECSecurityActionName is the user friendly name of this object.

```
NAME  IPSECSecurityActionName
DESC  The user friendly name of this entry.
EQUALITY  caseExactIA5Match
SYNTAX  IA5String
SINGLE-VALUED
```

The SecurityAction attribute states the security action for the flow.

```
NAME  SecurityAction
DESC  Security action for the datagram
```

```
EQUALITY  caseExactStringMatch
SYNTAX IA5String
SINGLE-VALUED
FORMAT The following values are allowed
        Permit
        Deny
```

```
        PermitIfInboundIPSec
SEMANTICS Deny means that the packet must be dropped.

           Permit means that the packet must be allowed and further
           processing depends on the presence of the IPSecProposalRef
           attribute. If such an attribute is present, then the packet
           must be secured by IPSec; else the packet is transmitted
           in the clear.

           PermitIfInboundIPSec means that if the packet has already
           received inbound IPSec processing, then it must be processed
           according to `Permit' rules; else it must be dropped. This
           is to disallow packets that attempt to bypass inbound
           IPSec processing.
```

The next two attributes specifies the two end points of the IPSec security association that must carry the traffic. These attributes are relevant if the SecurityAction attribute is `Permit' or `PermitIfInboundIPSec' and there is an IPSecProposalRef attribute implying that the traffic must be secured by IPSec.

For some applications, it may not be required to specify these two attributes and the defaults may suffice (see examples in section 8)

```
NAME  LocalIPSecTunnelEndPoint
DESC  Address of the local IPSec host
EQUALITY caseIgnoreMatch
SYNTAX IA5 String
SINGLE-VALUED
FORMAT The following formats are supported
        1:<IPv4address>
        2:<Host FQDN>
DEFAULT Any one of the local interface addresses for the host for
         which the policy is applicable

NAME  RemoteIPSecTunnelEndPoint
DESC  A list of potential addresses of the remote IPSec host
EQUALITY  caseIgnoreMatch
SYNTAX  IA5 String
MULTI-VALUED
FORMAT Same as LocalIPSecTunnelEndPoint
DEFAULT If the packet is a locally destined IPSec Quick Mode packet
         then the RemoteIPSecTunnelEndPoint is the source address in
         the packet (that matches the policy conditions)
```

If the packet is a data packet that is to be forwarded after IPSec processing then the RemoteIPSecTunnelEndPoint is the destination address in the packet (that matches the policy conditions)

SEMANTICS If the SecurityAction is Permit and there is an IPSecProposalRef attribute then, the flow described in the linked PolicyCondition object must be carried by an IPSec security association between the two hosts described by the LocalIPSecTunnelEndPoint and RemoteIPSecTunnelEndPoint attributes.

The LocalIPSecTunnelEndPoint attribute represents a particular interface for the local host. This is useful for multihomed hosts.

Multiple RemoteIPSecTunnelEndPoints are treated as logical OR.

The following six attributes together define the traffic in the Identity payload in the IPSec Quick Mode negotiation.

The LocalProxiedAddressRange, ProxiedIPProtocol and LocalProxiedPort attributes define the traffic for which the LocalIPSecTunnelEndPoint host acts as a proxy.

The RemoteProxiedAddressRange, ProxiedIPProtocol and RemoteProxiedPort attributes define the traffic for which the RemoteIPSecTunnelEndPoint host acts as a proxy.

The ProxiedHostScope attribute describes whether a separate IPSec Security Association is required for each pair of hosts in (LocalProxiedAddressRange, RemoteProxiedAddressRange) or only one is required for that entire range of hosts.

```
NAME  LocalProxiedAddressRange
DESC  Local proxied address range for use in ISAKMP Quick Mode payload
EQUALITY  caseIgnoreMatch
SYNTAX  IA5 String
SINGLE-VALUED
FORMAT identical to SourceIPAddressRange in the IPPolicyCondition class.
DEFAULT The entire address range


NAME  RemoteProxiedAddressRange
DESC  Remote proxied address range for use ISAKMP Quick Mode Identity
       payload
EQUALITY  caseIgnoreMatch
SYNTAX IA5 String
SINGLE-VALUED
FORMAT  identical to SourceIPAddressRange in the IPPolicyCondition class.
DEFAULT The entire address range
```

NAME ProxiedProtocol
DESC Proxied protocol for use in ISAKMP Quick Mode payload
EQUALITY caseIgnoreMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT The protocol value in the packet that matches the flow described
in the linked PolicyCondition object

```
NAME LocalProxiedPort
DESC local proxied port for use in ISAKMP Quick Mode Identity payload
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT The local port number in the packet that matches the flow.


NAME RemoteProxiedPort
DESC remote proxied port for use in ISAKMP Quick Mode Identity payload
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT The remote port number in the packet that matches the flow


NAME  ProxiedHostScope
DESC  Describes whether IPSec Security Association is one for each pair
       of hosts in (LocalProxiedAddressRange,RemoteProxiedAddressRange) or
       one for the entire range of hosts.
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The following values are allowed
        0x00
        0x01 (i.e. Least Significant Bit(LSB) is set)
        0x02 (i.e. LSB+1 is set)
        0x03 (i.e. both LSB and LSB+1 are set)
SEMANTICS LSB corresponds to local address while LSB+1 corresponds to
           Remote address. The semantics for each bit is identical.


           If LSB is reset then the entire set of addresses defined by
           the LocalProxiedAddressRange attribute must be carried over
           one IPSec security association.


           If the LSB is set then a distinct IPSec security association
           must be used for each host in the range of the
           LocalProxiedAddressRange attribute.


           Identical logic applies for the LSB+1 bit and the
           RemoteProxiedAddressRange attribute


DEFAULT The value 0x00; meaning that only one IPSec tunnel must be
         used for the entire set of LocalProxiedAddressRange and
```

RemoteAddressRange values.

The explicit rules for matching Proxied addresses are as follows:

1. If the packet is a locally destined IPSec Quick Mode packet (i.e. this host is acting as an IPSec Quick Mode responder), then the processing is as follows:

The source address in the packet must be contained in the RemoteTunnelEndPoint values (if specified).

If the LSB in ProxiedHostScope is set, then the IDci presented must be a single address within the RemoteProxiedAddresssRange and further, must be equal to the source address in the packet. Otherwise, the IDci must be entire RemoteProxiedAddressRange.

Similarly, if the LSB+1 bit is set then the IDcr presented must be a single address within the LocalProxiedAddressRange and further, must be equal to the destination address in the packet. Else, the IDcr presented must be the entire LocalProxiedAddressRange.

2. If the packet is one that is to be forwarded after IPSec processing, then the processing is to be done as follows.

   The source address in the received packet must belong to LocalProxiedAddressRange and the destination address in the received packet must belong to the RemoteProxiedAddressRange.

   If the LSB in ProxiedHostScope is set, then source address in the packet must be negotiated as IDci; otherwise the entire LocalProxiedAddressRange must be negotiated as IDci.

   If the LSB+1 bit in ProxiedHostScope is set, then destination address in the packet must be negotiated as IDcr; otherwise the entire RemoteProxiedAddressRange must be negotiated as IDcr.

As an example of a situation where two IPSec hosts must not negotiate the entire range of addresseses specified in the LocalProxiedAddressRange and RemoteProxiedAddressRange attributes, consider the remote access by users from a specific IPv4 subnet say 39.23.x.x. We might wish to say, for instance, that for any host attempting to do IPSec Quick Mode negotiation from the subnet 39.23.x.x, we require that the IDci presented comprise of the address of that host alone. We mandate this by specifying that the RemoteProxiedAddressRange is 39.23.x.x, but also that the ProxiedHostScope attribute value is 0x02 or 0x03. The meaning of these ProxiedHostScope values are described next and it implies that the source address in the received Quick Mode packet must be used to derive the IDci presented. This approach avoids having multiple IPSec actions, each containing single LocalProxiedAddressRange or

RemoteProxiedAddressRange values and provides flexibility in defining the traffic to be protected by an IPSec security association.

The IPSecProposalRef attribute contains a list of IPSec Proposals for the flow. Since LDAP does not support ordered lists, a composite string is required to define ordered IPSec proposals.

```
NAME  IPSecProposalRef
DESC  Ordered list of absolute DNs of of IPSecProposal objects
EQUALITY caseIgnoreMatch
SYNTAX    IA5String
MULTI-VALUED
FORMAT The format is `pref:IPSecProposalDN' where
         - pref is an integer denoting the relative preference of this
           proposal
         - IPSecProposalDN denotes the distinguishing name of an
           IPSecProposal object representing this proposal
```

Sometimes there can be multiple ISAKMPAction objects for the flow, e.g. if there are multiple ISAKMP security associations between the two IPSec hosts protecting this flow. In such scenarios, an ISAKMPActionRef attribute describes the particular ISAKMP security association that must carry this traffic.

```
NAME  ISAKMPActionRef
DESC  Absolute distinguised name of the ISAKMPAction object that
       describes the  ISAKMP action used to  carry the IPSec traffic
EQUALITY distinguishedNameMatch
SYNTAX  DN
SINGLE-VALUED
```

The rest of the attributes are as defined in Section 6.1 but apply to ISAKMP Quick Mode traffic.

7. Other classes

7.1. The class ISAKMPProposal

This class describes the attributes of an ISAKMP (phase one) proposal.

```
NAME  ISAKMPProposal
DESC  Describes ISAKMP proposal attributes
DERIVED FROM Top
AUXILIARY CLASSES NONE
MUST
      CommonName,
      ISAKMPAuthenticationMethod,
      ISAKMPHashAlgorithm,
      ISAKMPCipherAlgorithm,
MAY
      ISAKAMPProposalName,
```

```
      ISAKMPPrfAlgorithm,
      ISAKMPCipherKeyLength,
      ISAKMPCipherKeyRounds,
      DefaultDiffieHellmanGroupId,
```

```
      PrivateDiffieHellmanGroupRef,
      SecurityAssociationLifetimeSec,
      SecurityAssociationLifetimeKBytes
```

The ISAKMPProposalName defines the user friendly name of this entry.

```
NAME  ISAKMPProposalName
DESC  The user friendly name of this entry. The Name related  attributes are
       only for ease of user administration
EQUALITY  caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
```

The ISAKMPAuthenticationMethod attribute defines the ISAKMP/Oakley authentication method.

```
NAME  ISAKMPAuthenticationMethod
DESC  Authentication method for key exchange in ISAKMP
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT The acceptable values for are given in [4]
DEFAULT Implementation dependent
```

```
NAME  ISAKMPHashAlgorithm
DESC  Hash Algorithms for use in ISAKMP
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT The acceptable values for are given in [4]
DEFAULT Implementation dependent
```

```
NAME  ISAKMPCipherAlgorithm
DESC  Cipher Algorithms for use in ISAKMP
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT The acceptable values for are given in [4]
DEFAULT Implementation dependent
```

```
NAME  ISAKMPPRFAlgorithm
DESC  PseudoRandom function algorithm for use in ISAKMP
EQUALITY  integerMatch
```

SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values for are given in [4]
DEFAULT The value corresponding to HMAC

The following two attributes are related to some special ISAKMP ciphers.

```
NAME  ISAKMPCipherKeyLength
DESC  Keylength for use when ISAKMP Cipher algorithms are CAST, RC5 or
       Blowfish
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT Not applicable


NAME  ISAKMPCipherKeyRounds
DESC  Key rounds for use with some ISAKMP Cipher algorithms
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT Not applicable
```

ISAKMPCipherKeyRounds is not used at present, but may be needed for some new cipher algorithm.

DefaultDiffieHellmanGroupId attribute specifies the well known Diffie Hellman group Ids in case these are to be used. If on the other hand private groups are to be used, then the PrivateDiffieHellmanGroupRef provides a reference to the PrivateDiffieHellmanGroup object describing the group attributes.

```
NAME  DefaultDiffieHellmanGroupId
DESC  Default Diffie Hellman group ids: 1,2,3,4 defined in [4]
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT Implementation dependent


NAME  PrivateDiffieHellmanGroupRef
DESC  Absolute DN of an DiffieHellmanGroup object
EQUALITY  distinguishedNameMatch
SYNTAX  DN
SINGLE-VALUED
DEFAULT Not applicable
```

The following two attributes specify security association lifetimes.

```
NAME  SecurityAssociationLifetimeKBytes
DESC  Security Association Lifetime time in KBytes
EQUALITY  integerMatch
```

SYNTAX INTEGER
SINGLE-VALUED
DEFAULT Implementation dependent

NAME SecurityAssociationLifetimeSec
DESC Security Association Lifetime time in seconds
EQUALITY integerMatch

SYNTAX INTEGER
SINGLE-VALUED
DEFAULT Implementation dependent

7.2. The class IPSecProposal

This class describes an IPSec proposal for ISAKMP/Oakley Quick Mode negotiation. A proposal consists of combinations of AH, ESP and IPCOMP protocols.

The transform attributes of the AH protocol are specified by the AHProtocolTransformRef attribute that refers to an appropriate IPSecTransform object (described in section 7.3).

Similarly, the ESPProtocolTransformRef attribute specifies the transforms associated with the ESP protocol and the IPCOMPProtocolTransformRef attribute specifies the transforms associated with the IPCOMP protocol. The ESPProtocolTransformRef and IPCOMPProtocolTransformRef attribute refers to an appropriate IPSecTransform objects (described in section 7.3).

The attributes AHProtocolTransformRef, ESPProtocolTransformRef and IPCOMPProtocolTransformRef are all taken as logical ANDs when presented together. For example, when both an AHProtocolTransformRef and an ESPProtocolTransformRef are present, then both AH and ESP must be negotiated together.

The class definition is

```
NAME  IPSecProposal
DESC  Describes an IPSEC Proposal
DERIVED FROM  Top
MUST
      CommonName,
      PerfectForwardSecrecy
MAY
      IPSecProposalName,
      DefaultDiffieHellmanGroupId,
      PrivateDiffieHellmanGroupRef,
      AHProtocolTransformRef,
      ESPProtocolTransformRef,
      IPCOMPProtocolTransformRef
```

The attribute definitions are given below.

```
NAME  ISAKMPProposalName
DESC  The user friendly name of this entry.
EQUALITY  caseExactIA5Match
SYNTAX    IA5String
```

```
SINGLE-VALUED


    The PerfectForwardSecrecy attribute denotes whether a fresh Diffie
    Hellman Exchange is required in IPSec Quick Mode.  If this attribute
    value is 1 (i.e.  fresh Diffie Hellman exchange is required) then
    one of the Diffie Hellman attributes DefaultDiffieHellmanGroupId,
    PrivateDiffieHellmanGroupRef must be present in each of the referred
    IPSecTransform objects.


NAME  PerfectForwardSecrecy
DESC  Perfect forward secrecy requirement in IPSec Quick Mode
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
FORMAT The following values are defined
        1 (Required)
        0 (not required)


NAME  DefaultDiffieHellmanGroupId
DESC  Default Diffie Hellman group ids: 1,2,3,4 defined in [4]
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED


NAME  PrivateDiffieHellmanGroupRef
DESC  Absolute DN of a private DiffieHellmanGroup object
EQUALITY  distinguishedNameMatch
SYNTAX  DN
SINGLE-VALUED


    Note that the following reference object lists are defined as strings
    in order to emulate ordered lists which is currently not supported in
    LDAP.


NAME  AHProtocolTransformRef
DESC  Ordered list of absolute distinguished names of IPSecTransform objects
       corresponding to AH protocol
EQUALITY  caseIgnoreMatch
SYNTAX  IA5 String
MULTI-VALUED
FORMAT The format is `pref:IPSecTransformDN' where
         - pref is an integer denoting the relative preference of
           the transform. Lower number is higher preference.
         - IPSecTransformDN denotes the distinguishing name of an
```

IPSecTransform object corresponding to the AH protocol

NAME ESPProtocolTransformRef
DESC Ordered list of absolute distinguished names of IPSecTransform objects
corresponding to ESP protocol
EQUALITY caseIgnoreMatch

```
SYNTAX  IA5 String
MULTI-VALUED
FORMAT The format is `pref:IPSecTransformDN' where
       - pref is an integer denoting the relative preference of
         the transform. Lower number is higher preference.
       - IPSecTransformDN denotes the distinguishing name of an
         IPSecTransform object corresponding to the ESP protocol


NAME  IPCOMPProtocolTransformRef
DESC  Ordered list of absolute distinguished names of IPSecTransform objects
       corresponding to IPCOMP protocol
EQUALITY distinguishedNameMatch
SYNTAX  DN
MULTI-VALUED
FORMAT The format is `pref:IPSecTransformDN' where
       - pref is an integer denoting the relative preference of the
         transform. Lower number is higher preference.
       - IPSecTransformDN denotes the distinguishing name of an
         IPSecTransform object corresponding to the IPCOMP protocol
```

7.3. The class IPSecTransform

This class describes the attributes of an IPSec Quick Mode transform.

```
NAME  IPSecTransform
DESC  Describes IPSec transform attributes
DERIVED FROM  Top
MUST
      CommonName
      IPSecProtocolId
MAY
      IPSecTransformName,
      AHIntegrityAlgorithm,
      ESPIntegrityAlgorithm,
      ESPCipherAlgorithm,
      ESPCipherKeyLength,
      ESPCipherKeyRounds,
      IPCOMPCompressAlgorithm,
      IPCOMPVendorCompressAlgorithm,
      EncapsulationMode,
      SecurityAssociationLifetimeSec,
      SecurityAssociationLifetimeKBytes


NAME  ISAKMPTransformName
DESC  The user friendly name of this entry.The Name related  attributes are
       only for ease of user administration.
```

```
EQUALITY  caseExactIA5Match
SYNTAX    IA5String
SINGLE-VALUED
```

The IPSecProtocolId attribute denotes the IPSec protocol (e.g. AH, ESP, IPCOMP) corresponding to this transform object. For example, if the transform object denotes an AH`MD5 transform then the IPSecProtocolId is IPSEC`AH.

NAME IPSecProtocolId
DESC IPSec protocol corresponding to this transform
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values are given in [4].

The AHIntegrityAlgorithm and ESPIntegrityAlgorithm attributes denote the integrity transform (e.g. MD5, SHA etc.) in AH and ESP protocols respectively.

NAME AHIntegrityAlgorithm
DESC Integrity Algorithm for use in AH
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values are given in [4].
DEFAULT Not applicable

NAME ESPIntegrityAlgorithm
DESC Integrity Algorithm for use in ESP
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values are given in [4].
DEFAULT Not applicable

The EncapsulationMode describes the Tunnel or transport encapsulation mode.

NAME EncapsulationMode
DESC Encapsulation Mode: Tunnel or Transport
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values for in [4].
DEFAULT: the integer value corresponding to the Transport Mode

The ESPCipherAlgorithm attribute denotes the integrity transform (e.g. 3DES, IDEA etc.) in ESP.

NAME ESPCipherAlgorithm
DESC Cipher Algorithms for use in ESP
EQUALITY integerMatch
SYNTAX INTEGER

SINGLE-VALUED
FORMAT The acceptable values are given in [4]
DEFAULT Not applicable

NAME ESPCipherKeyLength
DESC Keylength for use when ESP Cipher algorithms are CAST, RC5 or Blowfish
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT Not applicable

NAME ESPCipherKeyRounds
DESC Key rounds for use with some ESP Cipher algorithms
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT Not applicable

ESPCipherKeyRounds is not used at present, but may be needed for some new cipher algorithm.

NAME IPCOMPCompressAlgorithm
DESC Compression Algorithms for use in IPCOMP
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
FORMAT The acceptable values are given in [4]
DEFAULT Implementation dependent

NAME IPCOMPVendorCompressAlgorithm
DESC Vendor specific Compression Algorithms for use in IPCOMP
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED
DEFAULT Not applicable

The VendorCompressAlgorithm attribute must be present when CompressAlgorithm represents OUI.

The following two attributes specify security association lifetimes. If a proposal consists of multiple protocols such as AH and ESP, then the lifetime values applies to each protocol as they are negotiated together.

NAME SecurityAssociationLifetimeKBytes
DESC Security Association Lifetime in KBytes
EQUALITY integerMatch
SYNTAX INTEGER
SINGLE-VALUED

```
DEFAULT Implementation dependent


NAME  SecurityAssociationLifetimeSec
DESC  Security Association Lifetime in seconds
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
DEFAULT Implementation dependent
```

7.4. The class PrivateDiffieHellmanGroup

This class defines a private Diffie Hellman Group.

```
NAME  PrivateDiffieHellmanGroup
DESC  Describes a private Diffie Hellman group attributes
DERIVED FROM Top
MUST
      CommonName
      DHGroupType
MAY
      PrivateDHGroupName,
      DHPrime,
      DHGenerator,
      DHGenerator1,
      DHGenerator2,
      DHCurveA,
      DHCurveB,
      DHFieldSize,
      DHOrder
```

The attribute definitions are as follows.

```
NAME  DHGroupType
DESC  The diffie Hellman group type for a DHGroup object:
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
SEMANTICS The acceptable values are given in [4]


NAME  DHFieldSize
DESC  GF size for elliptic curve groups
EQUALITY  integerMatch
SYNTAX  INTEGER
SINGLE-VALUED
```

NAME DHGenerator
DESC Group Generator
EQUALITY caseIgnoreMatch
SYNTAX IA5 String

SINGLE-VALUED

NAME DHCurveA
DESC Group Curve A for elliptic curve groups
EQUALITY caseIgnoreMatch
SYNTAX IA5 String
SINGLE-VALUED

NAME DHCurveB
DESC Group Curve B for elliptic curve groups
EQUALITY caseIgnoreMatch
SYNTAX IA5 String
SINGLE-VALUED

NAME DHOrder
DESC Group Order for elliptic curve groups
EQUALITY caseIgnoreMatch
SYNTAX IA5 String
SINGLE-VALUED

8. VPN Schema Usage Examples

In this section we describe some usage scenarios for VPNs. The intent is not to be very complete in specifying all the attributes, rather to show how the important attributes are to be defined. The objects are all written in LDIF notation.

8.1. Scenario I: Intranet communication

```
  - - - - - - - - - - - - - - - - - - - - - - - - - -
 |                                                    |
 |                                                    |
 S1, TCP, any port  ---------------------> S2, TCP, port 8000-8080
 |                                                    |
 |                      Intranet                      |
 - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The requirements are:

- All hosts on subnet S1 must use IPSec to communicate to hosts on subnet S2 and (HTTP) ports 8000-8080

- Only hosts on subnet S1 are allowed to initiate connections

- No intermediate gateways are required

8.1.1. ISAKMP rules for each host in S1 and S2

Each host in S1 and S2 needs to have the following rule.

```
dn: cn=S1-S2-isakmp-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: ISAKMP
PolicyType:  ISAKMP
PolicyConditionRef: cn=S1-S2-isakmp-traffic,o=XYZ, c=US,
PolicyActionRef: cn=S1-S2-isakmp-action, o=XYZ, c=US


dn: cn=S1-S2-isakmp-traffic, o=XYZ, c=US
Objectclass: IPPolicyCondition
SourceAddressRange: S1
DestinationAddressRange: S2
IPProtocolRange: 17 (i.e. UDP)
SourcePortRange: 500 (i.e. ISAKMP port)
DestinationPortRange: 500 (i.e. ISAKMP port)


dn: cn=S1-S2-isakmp-action, o=XYZ, c=US
Objectclass: ISAKMPAction
ISAKMPProposalRef: cn=S1-S2-isakmp-proposal,o=XYZ, c=US


dn: cn=S1-S2-isakmp-proposal, o=XYZ, c=US
Objectclass: ISAKAMPProposal
ISAKMPHashAlgorithm: 2(i.e. SHA)
ISAKMPAuthenticationMethod: 4 (i.e. RSA encryption)
ISAKMPCipherAlgorithm: 5(i.e. 3DES)
SecurityAssociationLifetimeSec: 3600
```

Note that there must be no IPPolicyCondition object with S2 as the source address range and S1 as the destination address range, since hosts in S2 are not allowed to initiate ISAKMP negotiations.

8.1.2. IPSec Rules for each host in S1

For the sake of illustration suppose that the following two IPSec proposals need to be negotiated.

- the first (preferred) proposal consists of only ESP protocol with 3DES as cipher and SHA as the integrity algorithm,

- the second proposal consists of both AH and ESP protocols; SHA is the integrity algorithm for AH while 3DES is the cipher for ESP. There is no integrity algorithm for ESP in this proposal.

Three IPSec rules are needed for hosts on subnet S1::

1. one rule for handling data packets from S2 to S1: this states that such packets must arrive at S1 within an IPSec security association. Because of this rule, it would not be possible to send non-IPSec packets from S2 to S1 on src port 8000-8080.

```
dn: cn=S2-S1-HTTP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecDataRemote
PolicyConditionRef: cn=S2-S1-HTTP-traffic, o=XYZ, c=US
PolicyActionRef: cn=inboundIPSecAction, o=XYZ,c=US


dn: cn=S2-S1-HTTP-traffic, o=XYZ, c=US
Objectclass: IPPolicyCondition
SourceIPAddressRange: S2
DestinationIPAddressRange: S1
SourcePortRange: 8000:8080
IPProtocolRange: 4 (i.e. TCP)


dn: cn= inboundIPSecIPSecAction, o=XYZ, c=US
Objectclass: IPSecSecurityAction
SecurityAction: PermitIfInboundIPSec
```

2. one rule for data packets from S1 to S2: this states that such packets must be secured by IPSec processing.

```
dn: cn= S1-S2-HTTP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecDataLocal
PolicyConditionRef: cn=S1-S2-HTTP-traffic, o=XYZ, c=US
PolicyActionRef: cn=S1-S2-HTTP-IPSec-action, o=XYZ,c=US


dn: cn=S1-S2-HTTP-traffic, o=XYZ, c=US
Objectclass: IPPolicyCondition
SourceIPAddressRange: S1
DestinationIPAddressRange: S2
DestinationPortRange: 8000:8080
IPProtocolRange: 4 (i.e. TCP)


dn: cn= S1-S2-HTTP-IPSec-action, o=XYZ, c=US,
Objectclass: IPSecSecurityAction
SecurityAction: Permit
```

```
LocalProxiedAddressRange: S1
RemoteProxiedAddressRange: S2
LocalProxiedPort: 0
RemoteProxiedPort: 8000 : 8080
ProxiedProtocol: 4
ProxiedHostScope: 0x11
IPSecProposalRef: 1: cn= ESPProposal, o=XYZ, c=US,
```

```
                IPSecProposalRef: 2: cn= AHESPProposal, o=XYZ, c=US


                dn: cn=ESPProposal,o=XYZ, c=US
                Objectclass: IPSecProposal
                ESPProtocolTransformRef: 1: cn= AuthEncryptTransform,o=XYZ,
c=US


                dn: cn=AHESPProposal, o=XYZ, c=US
                Objectclass: IPSecProposal
                AHProtocolTransformRef: 1: cn= AuthTransform, o=XYZ, c=US
                ESPProtocolTransformRef: 1: cn= EncryptTransform, o=XYZ, c=US


                dn: cn= AuthEncryptTransform,o=XYZ, c=US,
                Objectclass: IPSecTransform
                IPSecProtocolId: 3 (i.e. IPSEC`PROTO`ESP)
                ESPCipherAlgorithm: 3 (i.e. 3DES)
                ESPIntegrityAlgorithm: 2 (i.e. HMAC-SHA-1)
                EncapsulationMode: 2 (i.e. transport)
                SecurityAssociationLifetimeSec: 3600


                dn: cn= AuthTransform,o=XYZ, c=US
                Objectclass: IPSecTransform
                IPSecProtocolId: 2 (i.e. IPSEC`PROTO`AH)
                AHIntegrityAlgorithm: 2 (i.e. HMAC-SHA-1)
                EncapsulationMode: 1 (i.e. tunnel)
                SecurityAssociationLifetimeSec: 3600


                dn:  cn= EncryptTransform, o=XYZ, c=US
                Objectclass: IPSecTransform
                IPSecProtocolId: 3 (i.e. IPSEC`PROTO`ESP)
                ESPCipherAlgorithm: 3 (i.e. 3DES)
                EncapsulationMode: 2 (i.e. transport)
                SecurityAssociationLifetimeSec: 3600
```

3. one for IPSec packets from S1 to S2 (that is, packets with protocol field AH or ESP). This would state whether S1 and S2 can communicate directly or a gateway has to be traversed.

```
                dn: cn= S1-S2-AHESP-rule, o=XYZ, c=US
                Objectclass: Policy
                PolicyType: IPSecDataLocal
                PolicyConditionRef: cn=S1-S2-AHESP-traffic, o=XYZ, c=US
                PolicyActionRef: cn=clearIPSecSecurityAction, o=XYZ, c=US
```

```
dn: cn=S1-S2-AHESP-traffic, o=XYZ, c=US,
Objectclass: IPPolicyCondition
SourceIPAddressRange: S1
DestinationIPAddressRange: S2
IPProtocolRange: 50-51 (i.e. AH and ESP)


dn: cn=clearIPSecSecurityAction, o=XYZ, c=US
```

```
Objectclass: IPSecSecurityAction
SecurityAction: Permit
```

8.1.3. IPSec Rules for each host in S2

The situation for hosts in S2 is symmetric to those for S1, except that a policy is needed for hosts in S2 to respond to ISAKMP Quick Mode negotiations. Hosts in S1 do not need such a policy since they only initiate ISAKMP.

Such a policy is needed since the packet header in ISAKMP Quick Mode is different than in a data packet and we want to make it straightforward for hosts to match policies.

Hence for hosts in S2, the following IPSec rules are needed:

- Three rules as described in section 8.1.2 with the difference that source and destination addresses, port numbers etc. must be reversed.

- An extra rule that enables hosts on S2 to respond to ISAKMP Phase 2 signalling.

```
dn: cn=S1-S2-isakmp-QuickMode-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecPhase2
PolicyConditionRef: cn=S1-S2-isakmp-traffic,o=XYZ, c=US,
PolicyActionRef: cn=S2-HTTP-S1-ipsec-action, o=XYZ, c=US


dn: cn= S2-HTTP-S1-IPSec-action, o=XYZ, c=US,
Objectclass: IPSecSecurityAction
SecurityAction: Permit
LocalProxiedAddressRange: S2
RemoteProxiedAddressRange: S1
LocalProxiedPort: 8000 : 8080
RemoteProxiedPort: 0
ProxiedProtocol: 4
ProxiedHostScope: 0x11
IPSecProposalRef: 1: cn= ESPProposal, o=XYZ, c=US,
IPSecProposalRef: 2: cn= AHESPProposal, o=XYZ, c=US
```

The IPSecProposal objects are defined in section 8.1.2.

8.2. Scenario II: Remote access to intranet via an ISP

This case differs from the previous in that subnet S2 is behind a security gateway GW2. The traffic between subnets S1 and S2 on destination port range 8000-8080 must be sent within an per host IPSec tunnel between an host on S1 and GW2.

```
                           ----------------------------------
                          |                                  |
 S1,TCP ---Internet---  GW2---Intranet-------->S2,TCP,HTTP ports

     <-----------------end-to-end IPSec------------>
     <---outer IPSec --->|                                  |
          tunnel          ----------------------------------
```

8.2.1. Rules for each host in S1

Identical to those in section 8.1 since from S2's point of view, nothing has changed.

8.2.2. Rules for each host in S2

ISAKMP Rules:

An additional rule is required for communication between hosts on S1 and GW2. Typically the traffic profile described in the PolicyCondition object for S1-S2 rule will be broad enough to include the S1 and GW2. If this is not the case then a new rule has to be added as in section 8.1.1 by replacing the subnet S2 with the gateway GW2.

IPSec rules:

The difference between this case and the intranet case in section 8.1.2 is that hosts on S1 now have to send S1-S2 traffic via the gateway GW2.

To accomplish this, simply replace the rule whose DN equals ``cn= S1-S2-AHESP-rule, o=XYZ, c=US'' in section 8.1.2 by the following two rules: (Note that objects not defined here are defined earlier in this section)

1. One rule which states that IPSec packets between S1 and S2 must be sent within an IPSec tunnel between S1 and GW2.

```
dn: cn= S1-S2-AHESP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
```

PolicyType: IPSecDataLocal
PolicyConditionRef: cn=S1-S2-AHESP-traffic, o=XYZ, c=US
PolicyActionRef: cn=AHTunnelSecurityAction, o=XYZ, c=US

```
dn: cn=AHTunnelSecurityAction, o=XYZ, c=US
Objectclass: IPSecSecurityAction
SecurityAction: Permit
RemoteIPSecTunnelEndPoint: GW2
LocalProxiedAddressRange: S1
RemoteProxiedAddressRange: S2
ProxiedProtocol: 0
LocalProxiedPort:0
RemoteProxiedPort:0
ProxiedHostScope: 0x11
IPSecProposalRef: cn=AuthTunnelProposal, o=XYZ, c=US


dn:  cn= AuthTunnelProposal,o=XYZ, c=US
Objectclass: IPSecProposal
IPSecTransformRef: 1: cn= AuthTransform, o=XYZ, c=US
```

2. one rule that states that hosts on S1 and GW2 need not traverse any intermediate gateways.

```
dn: cn=S1-GW2-AHESP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecDataLocal
TrafficProfileRef: cn=S1-GW2-AHESP-traffic, o=XYZ, c=US
PolicyActionRefe: cn=clearIPSecSecurityAction, o=XYZ,c=US


dn: cn=S1-GW2-AHESP-traffic, o=XYZ, c=US
Objectclass: PolicyCondition
SourceAddressRange: S1
DestinationAddressRange: GW2
IPProtocolRange: 50:51
```

8.2.3. Rules for GW2

Only the IPSec rules are described here. The ISAKMP rule between GW2 and hosts on S1 can be generated easily. Note that objects not defined here are defined earlier in this section.

1. A rule that states that packets from S1 to S2 on destination port 8000-8080 must be received inside of an IPSec security association, and then must be sent out in the clear.

```
dn: cn= S1-S2-GatewayRemoteAccessRule, o=XYZ, c=US
Objectclass: Policy
```

PolicyScope: IPSec
PolicyType: IPSecDataRemote
TrafficProfileRef: cn=S1-S2-HTTP-traffic, o=XYZ, c=US
PolicyActionRef: cn=S1-GW2-inbound-SecurityAction, o=XYZ,c=US

```
dn: cn=S1-GW2-inbound-SecurityAction, o=XYZ, c=US
Objectclass: IPSecSecurityAction
SecurityAction:  PermitIfInboundIPSec
```

2. A rule that states that packets from S2 to S1 on source port 8000 to 8080 must be secured by ipsec on the outbound path.

```
dn: cn= S2-S1-GatewayRemoteAccessRule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecDataLocal
TrafficProfileRef: cn=S2-HTTP-S1-traffic, o=XYZ, c=US
PolicyActionRef: cn=GW2-S1-HTTP-SecurityAction, o=XYZ, c=US
```

```
dn: cn=GW2-S1-HTTP-SecurityAction, o=XYZ, c=US
Objectclass: IPSecSecurityAction
SecurityAction: Permit
LocalIPSecTunnelEndpoint: GW2
LocalProxiedAddressRange: S2
RemoteProxiedAddressRange: S1
ProxiedProtocol: 0
LocalProxiedPort:0
RemoteProxiedPort:0
ProxiedHostScope: 0x11
ProxiedProtocol: 4(i.e. TCP)
IPSecProposalRef: cn=AHTunnelProposal, o=XYZ, c=US
```

3. A rule that states that GW2 and hosts on S1 can communicate directly.

```
dn: cn=GW2-S1-AHESP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: ISAKMPDataLocal
TrafficProfileRef: cn=GW2-S1-EHESP-traffic, o=XYZ, c=US
PolicyActionRef: cn=clearIPSecSecurityAction, o=XYZ, c=US
```

```
dn: cn=GW2-S1-AHESP-traffic, o=XYZ, c=US
Objectclass: PolicyCondition
SourceAddressRange: GW2
DestinationAddressRange: S1
IPProtocolRange: 50-51 (i.e. AH and ESP)
```

4. A rule for GW2 to respond to ISAKMP Quick Mode packets from hosts in S1.

```
dn: cn=S1-GW2-isakmp-QuickMode-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType:  ISAKMPPhase2
```

```
                PolicyConditionRef: cn=S1-GW2-isakmp-traffic,o=XYZ, c=US,
                PolicyActionRef: cn=GW2-S1-HTTP-SecurityAction, o=XYZ, c=US


                dn: cn=S1-GW2-isakmp-traffic, o=XYZ, c=US
                Objectclass: IPPolicyCondition
                SourceAddressRange: S1
                DestinationAddressRange: GW2
                IPProtocolRange: 17 (i.e. UDP)
                SourcePortRange: 500 (i.e. ISAKMP port)
                DestinationPortRange: 500 (i.e. ISAKMP port)
```

8.3. Scenario III: Corporate Branch office to Main office

Suppose that hosts on subnets S1 and S2 are not IPSec enabled. therefore traffic initiated by any host on subnet S1 and destined to any host subnet S2 and port 80 is to be carried by the security gateways GW1 and GW2 within an IPSec security association in tunnel mode as show below.

```
       ----------------                   ---------------------
     |                |                   |                   |
           Intranet                             Intranet
   S1,TCP,-----------GW1-----Internet---GW2---------->S2,TCP,
    Any port         <-----IPSec ------>          port 8000-8080
                           AH tunnel
     |                |                   |                   |
      ---------------                     -------------------
```

Rules for GW1 are described here since those for GW2 are completely symmetric except the ISAKMP Quick Mode responder rule. Also, only IPSec rules are described since ISAKMP rules are straightforward. Note that objects not defined here are defined earlier in this section.

1. The first rule for the gateway GW1 concerns packets received from hosts on subnet S1 destined to hosts on subnet S2 and on port 8000-8080. These packets must be sent to GW2 within ONE IPSec tunnel. Note the use of the ProxiedHostScope attribute.

```
               dn: cn= S1-S2-BrOffRule, o=XYZ, c=US
               Objectclass: Policy
               PolicyScope: IPSec
               PolicyType: IPSecDataLocal
               TrafficProfileRef: cn=S1-S2-HTTP-traffic, o=XYZ, c=US
               PolicyActionRef: cn=S1-S2-BrOffSecAction, o=XYZ,c=US
```

```
dn: cn=S1-S2-BrOffSecAction, o=XYZ, c=US
Objectclass: IPSecSecurityAction
SecurityAction: Permit
```

```
LocalIPSecTunnelEndPoint: GW1
RemoteIPSecTunnelEndPoint: GW2
LocalProxiedAddressRange: S1
RemoteProxiedAddressRange: S2
LocalProxiedPort: 0
RemoteProxiedPort: 8000 : 8080
ProxiedProtocol: 4
ProxiedHostScope: 0x00
IPSecProposalRef: cn=AHTunnelProposal, o=XYZ, c=US
```

2. The second rule states that GW1 and GW2 can communicate directly without any intermediate gateways.

```
dn: cn=GW1-GW2-AHESP-rule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: IPSecDataLocal
TrafficProfileRef: cn=GW1-GW2-AHESP-traffic, o=XYZ, c=US
PolicyActionRef: cn=clearIPSecSecurityAction, o=XYZ, c=US
```

```
dn: cn=GW1-GW2-AHESP-traffic, o=XYZ, c=US
Objectclass: PolicyCondition
SourceIPAddressRange: GW1
DestinationIPAddressRange: GW2
IPProtocolRange: 50-51 (i.e. AH and ESP)
```

3. The third rule states that packets from S2 to S1 must receive inbound IPSec processing and then forwarded in the clear.

```
dn: cn= S2-S1-BrOffRule, o=XYZ, c=US
Objectclass: Policy
PolicyScope: IPSec
PolicyType: PolicyDataRemote
PolicyConditionRef: cn=S2-S1-HTTP-traffic, o=XYZ, c=US
PolicyActionRef: cn=inboundIPSecAction, o=XYZ,c=US
```

9. Security Considerations

This draft presents a policy model of the IPSec documents. All security considerations within those actual specification MUST be considered prior to implementing a policy architecture.

References

[1] R. Atkinson, ``Security Architecture for the Internet Protocol'',

draft-ietf-ipsec-arch-sec-01

[2] D. Maughan, M. Schertler, M. Schneider, J. Turner, ``
Internet Security Association and Key Management'',
draft-ietf-ipsec-isakmp-09

[3] M. Wahl, T. Howes, S. Kille, ``Lightweight Directory Access
Protocol (v3)'', RFC 2251

[4] D. Harkins, ``The Internet Key Exchange'', draft-ietf-ipsec-isakmp-
oakl*
*ey-06

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

[6] R. Rajan, J-C. Martin, S. Kamat, M. See and R. Chaudhury,
``Schema for Differentiated Services and Integrated Services in
Networks'', draft-ietf-policy-qosschema-00.txt

[7] S. Judd and J. Strassner, ``Directory Enabled Networks -
Information Model and Base Schema'' - Draft v3.0r5 DEN
Specifications, September 1998

[8] Common Information Model (CIM) Specification, Desktop Management
Task Force, Version 2.0, Mar. 1998.

[9] R. Pereira and P. Bhattacharya, ``IPSec Policy Data Model'',
draft-ietf-ipsec-policy-model-00.txt

Acknowledgments

The IBM authors would like to thank Pau Cheng, Will Fiveash,
Skip Booth and Charlie Kunzinger for many useful discussions and
suggestions.

Contact Address

Partha P Bhattacharya
Phone: (914) 784-7981
Email: partha@watson.ibm.com
IBM T. J. Watson Research Center
Rob Adams
Phone: (425) 558-2285
Email: radams@cisco.com

Cisco Systems
Roy Pereira
Phone: (613) 599-3610 x 4808

Email: rpereira@timestep.com
TimeStep Corporation

William Dixon
Phone: (425) 703-8729
Email: wdixon@microsoft.com
Microsoft Corporation

Raju Rajan
Phone: (914) 784-7260
Email: raju@watson.ibm.com
IBM T. J. Watson Research Center