

Network Working Group
Request for Comments: 4807
Category: Standards Track

M. Baer
Sparta, Inc.
R. Charlet
Self
W. Hardaker
Sparta, Inc.
R. Story
Revelstone Software
C. Wang
ARO
March 2007

IPsec Security Policy Database Configuration MIB

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The IETF Trust (2007).

Abstract

This document defines a Structure of Management Information Version 2 (SMIv2) Management Information Base (MIB) module for configuring the security policy database of a device implementing the IPsec protocol. The policy-based packet filtering and the corresponding execution of actions described in this document are of a more general nature than for IPsec configuration alone, such as for configuration of a firewall. This MIB module is designed to be extensible with other enterprise or standards-based defined packet filters and actions.

Table of Contents

1. Introduction	3
2. Terminology	3
3. The Internet-Standard Management Framework	3
4. Relationship to the DMTF Policy Model	3
5. MIB Module Overview	4
5.1. Usage Tutorial	6
5.1.1. Notational Conventions	6
5.1.2. Implementing an Example SPD Policy	7
6. MIB Definition	8
7. Security Considerations	65
7.1. Introduction	65
7.2. Protecting against Unauthenticated Access	66
7.3. Protecting against Involuntary Disclosure	66
7.4. Bootstrapping Your Configuration	67
8. IANA Considerations	67
9. Acknowledgments	68
10. References	68
10.1. Normative References	68
10.2. Informative References	69

1. Introduction

This document defines a MIB module for configuration of an IPsec security policy database (SPD). The IPsec model this MIB is designed to configure is based on the "IPsec Configuration Policy Model" (IPCP) [RFC3585]. The IPCP's IPsec model is, in turn, derived from the Distributed Management Task Force's (DMTF) IPsec model (see below) and from the IPsec model specified in RFC 2401 [RFC2401]. Note: RFC 2401 has been updated by RFC 4301 [RFC4301], but this implementation is based on RFC 2401. The policy-based packet filtering and the corresponding execution of actions configured by this MIB is of a more general nature than for IPsec configuration only, such as for configuration of a firewall. It is possible to extend this MIB module and add other packet-transforming actions that are performed conditionally on an interface's network traffic.

The IPsec- and IKE-specific actions are as documented in [IPsec-ACTION] and [IKE-ACTION], respectively, and are not documented in this document.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. The Internet-Standard Management Framework

For a detailed overview of the documents that describe the current Internet-Standard Management Framework, please refer to section 7 of RFC 3410 [RFC3410]

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIV2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

4. Relationship to the DMTF Policy Model

The Distributed Management Task Force (DMTF) has created an object oriented model of IPsec policy information known as the IPsec Policy Model White Paper [IPPMWP]. The "IPsec Configuration Policy Model" (IPCP) [RFC3585] is based, in large part, on the DMTF's IPsec policy model and on RFC 2401 [RFC2401]. The IPCP document describes a model

for configuring IPsec. This MIB module is a task-specific derivation (i.e., an SMIV2 instantiation) of the IPCP's IPsec configuration model for use with Simple Network Management Protocol version 3 (SNMPv3).

The high-level areas where this MIB module diverges from the IPCP model are:

- o Policies, Groups, Conditions, and some levels of Actions are generically named. In other words, IPsec-specific prefixes like "SA" (Security Association), or "IPsec", are not used. This naming convention is used because packet classification and the matching of conditions to actions is more general than IPsec. The tables in this document can possibly be reused by other packet-transforming actions, which need to conditionally act on packets matching filters.
- o Filters are implemented in a more generic and scalable manner, rather than enforcing the condition/filtering pairing of the IPCP and its restrictions upon the user. This MIB module offers a compound filter object providing greater flexibility for complex filters than the IPCP.

5. MIB Module Overview

The MIB module is modularized into several different parts: rules, filters, and actions.

The rules section associates endpoints and groups of rules, and consists of the `spdEndpointToGroupTable`, `spdGroupContentsTable`, and the `spdRuleDefinitionTable`. Each row of the `spdRuleDefinitionTable` connects a filter to an action. It should also be noted that by referencing the `spdCompoundFilterTable`, the `spdRuleDefinitionTable`'s filter column can indicate a set of filters to be processed. Likewise, by referencing the `spdCompoundActionTable`, the `spdRuleDefinitionTable`'s action column can indicate multiple actions to be executed.

This MIB is structured to allow for reuse through the future creation of extension tables that provide additional filters and/or actions. In fact, the companion documents to this one ([IPsec-ACTION] and [IKE-ACTION]) do just that and define IPsec- and IKE-specific actions to be used within this SPD configuration MIB. Note: it is expected that, in order to function properly, extension action MIBs may impose additional limitations on the objects in this MIB and how they can be used with the extended actions. An extension action may only support a subset of the configuration options available in this MIB.

The filter section of the MIB module is composed of the different types of filters in the Policy Model. It is made up of the `spdTrueFilter`, `spdCompoundFilterTable`, `spdSubfiltersTable`, `spdIpHeaderFilterTable`, `spdIpOffsetFilterTable`, `spdTimeFilterTable`, `spdIpsoHeaderFilterTable`.

The action section of this MIB module contains only the simple static actions required for the firewall processing that an IPsec SPD implementation requires (e.g., accept, drop, log, etc.). The companion documents of this document define the complex actions necessary for IPsec and IKE negotiations.

As may have been noticed above, the MIB uses recursion in a similar manner in several different places. In particular, the `spdGroupContentsTable`, the `spdCompoundFilterTable` / `spdSubfiltersTable` combination, and the `spdCompoundActionTable` / `spdSubactionsTable` combination can reference themselves.

In the case of the `spdGroupContentsTable`, a row can indicate a rule (i.e., a row in the `spdRuleDefinitionTable`) or a group (i.e., another set of one or more rows in the `spdGroupContentsTable`). This way, a group can contain a set of rules and sub-groups. Sub-groups are just other groups defined in the `spdGroupContentsTable`. There is no inherent MIB limit to the depth of nesting of groups.

The `spdCompoundFilterTable` / `spdSubfiltersTable` combination and `spdCompoundActionTable` / `spdSubactionsTable` combination are designed almost identically, with one being for filters and the other for actions, respectively. The following descriptions for the compound filter tables can be directly applied to the compound action tables.

The combination of the tables `spdCompoundFilterTable` and `spdSubfiltersTable` allow a user to create a set of filters that can be referenced from any table as a single filter. A row in the `spdCompoundFilterTable` has the basic configuration information for the compound filter. The index of `spdCompoundFilterTable`, `spdCompFiltName`, is also used as a partial index to reference a set of ordered rows in the `spdSubfiltersTable`. Each row in `spdSubfiltersTable` points to a row in another filter table. In this way, the set of rows in `spdSubfiltersTable` with a matching `spdCompFiltName`, together with the row in `spdCompoundFilterTable` indexed by `spdCompFiltName`, create a compound filter. Note that it is possible for a row in the `spdSubfiltersTable` to point to a row in the `spdCompoundFilterTable`. This recursion allows the creation of a filter set that includes other filter sets within it. There is no inherent MIB limit to the nesting of compound filters within compound filters.

5.1. Usage Tutorial

In order to use the tables contained in this document, a general understanding of firewall processing is helpful. The processing of the security policy database (SPD) involves applying a set of SPD rules to an interface on a device. The given set of rules to apply to any given interface is defined within the `spdEndpointToGroupTable` table. This table maps a given interface to a group of rules. In this table, the interface itself is specified using its assigned address. There is also one group of rules per direction (ingress and egress).

5.1.1. Notational Conventions

Notes about the following example operations:

1. All the example operations in the following section make use of default values for all columns not listed. The operations and column values given in the examples are the minimal SNMP Varbinds that must be sent to create a row.
2. The example operations are formatted such that a row (i.e., the table's Entry object) is operated on by using the indexes to that row and the column values for that row.
3. Below is a generic example of the notation used in the following section's examples of this MIB's usage. This example indicates that the MIB row to be set is the row with the index values of `value1` for `index1`, and `value2` for `index2`. Within this row, `column1` is set to `column_value1`, and `column2` is set to `column_value2`.

```
rowEntry(index1      = value1,
          index2      = value2)
      = (column1      = column_value1,
         column2      = column_value2)
```

4. The below is a specific example of the notation used in the following section's examples of this MIB's usage. This example represents the status column of a row in the `IP-MIB::ipAddressTable` table being set to deprecated. The index values for this row are IPv4 and 192.0.2.1. The example notation would look like the following:

```
ipAddressEntry(ipAddressAddrType = 1,           -- ipv4
                ipAddressAddr     = 0xC0000201 ) -- 192.0.2.1
      = (ipAddressStatus = 2)                  -- deprecated
```

5.1.2. Implementing an Example SPD Policy

As an example, let us define the following administrative policy: On the network interface with IP address 192.0.2.1, all traffic from host 192.0.2.6 will be dropped and all other traffic will be accepted.

This policy is enforced by setting the values in the MIB to do the following:

- o create a filter for 192.0.2.6
- o create a rule that connects the 192.0.2.6 filter to a packet drop action
- o create a rule that always accepts packets
- o group these rules together in the proper order so that the 192.0.2.6 drop rule is checked first.
- o connect this group of rules to the 192.0.2.1 interface

The first step to do this is creating the filter for the IPv4 address 192.0.2.6:

```
SpdIpHeaderFilterEntry(spdIpHeadFiltName = "192.0.2.6")
    = (spdIpHeadFiltType           = 0x80,           -- sourceAddress
       spdIpHeadFiltIPVersion      = 1,              -- IPv4
       spdIpHeadFiltSrcAddressBegin = 0xC0000206,     -- 192.0.2.6
       spdIpHeadFiltSrcAddressEnd   = 0xC0000206,     -- 192.0.2.6
       spdIpHeadFiltRowStatus       = 4)              -- createAndGo
```

Next, a rule is created to connect the above "192.0.2.6" filter to an action to "drop" the packet, as follows:

```
spdRuleDefinitionEntry(spdRuleDefName = "drop from 192.0.2.6")
    = (spdRuleDefFilter           =
       spdIpHeadFiltType.9.49.57.50.46.48.46.50.46.54,
       spdRuleDefAction           = spdDropAction.0,
       spdRuleDefRowStatus        = 4)              -- createAndGo
```

Next, a rule is created that accepts all packets:

```
spdRuleDefinitionEntry(spdRuleDefName = "accept all")
    = (spdRuleDefFilter           = spdTrueFilter.0,
       spdRuleDefAction           = spdAcceptAction.0,
       spdRuleDefRowStatus        = 4)              -- createAndGo
```

Next, these two rules are grouped together. Rule groups attached to an interface are processed one row at a time. The rows are processed from lowest to highest `spdGroupContPriority` value. Because the row that references the "accept all" rule should be processed last, it is given the higher `spdGroupContPriority` value.

```
SpdGroupContentsEntry(spdGroupContName      = "ingress",
                      spdGroupContPriority = 65535)
  = (spdGroupContComponentName = "accept all",
     spdGroupContRowStatus      = 4)      -- createAndGo
```

```
SpdGroupContentsEntry(spdGroupContName      = "ingress",
                      spdGroupContPriority = 1000)
  = (spdGroupContComponentName = "drop from 192.0.2.6",
     spdGroupContRowStatus      = 4)      -- createAndGo
```

Finally, this group of rules is connected to the 192.0.2.1 interface as follows:

```
SpdEndpointToGroupEntry(spdEndGroupDirection = 1,      -- ingress
                        spdEndGroupIdType     = 4,      -- IPv4
                        spdEndGroupAddress    = 0xC0000001)
  = (spdEndGroupName = "ingress",
     spdEndGroupRowStatus = 4)      -- createAndGo
```

This completes the necessary steps to implement the policy. Once all of these rules have been applied, the policy should take effect.

6. MIB Definition

The following MIB Module imports from: [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3289], [RFC3411], and [RFC4001]. It also uses definitions from [RFC1108], [RFC3060], and [RFC3629].

```
IPSEC-SPD-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
    MODULE-IDENTITY, OBJECT-TYPE, NOTIFICATION-TYPE, Integer32,
    Unsigned32, mib-2
        FROM SNMPv2-SMI
        -- [RFC2578]
```

```
    TEXTUAL-CONVENTION, RowStatus, TruthValue,
    TimeStamp, StorageType, VariablePointer
        FROM SNMPv2-TC
        -- [RFC2579]
```



```
MODULE-COMPLIANCE, OBJECT-GROUP, NOTIFICATION-GROUP
    FROM SNMPv2-CONF
    -- [RFC2580]
```

```
InterfaceIndex
    FROM IF-MIB
    -- [RFC2863]
```

```
diffServMIBMultiFieldClfrGroup, IfDirection,
diffServMultiFieldClfrNextFree
    FROM DIFFSERV-MIB
    -- [RFC3289]
```

```
InetAddressType, InetAddress
    FROM INET-ADDRESS-MIB
    -- [RFC4001]
```

```
SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
    -- [RFC3411]
```

```
;
```

```
--
-- module identity
--
```

```
spdMIB MODULE-IDENTITY
```

```
    LAST-UPDATED "200702070000Z"      -- 7 February 2007
    ORGANIZATION "IETF IP Security Policy Working Group"
    CONTACT-INFO "Michael Baer
        P.O. Box 72682
        Davis, CA 95617
        Phone: +1 530 902 3131
        Email: baerm@tislabs.com
```

```
    Ricky Charlet
    Email: rcharlet@alumni.calpoly.edu
```

```
    Wes Hardaker
    Sparta, Inc.
    P.O. Box 382
    Davis, CA 95617
    Phone: +1 530 792 1913
    Email: hardaker@tislabs.com
```

```
    Robert Story
    Revelstone Software
    PO Box 1812
```

Tucker, GA 30085
 Phone: +1 770 617 3722
 Email: rstory@ipsp.revelstone.com

Cliff Wang
 ARO
 4300 S. Miami Blvd.
 Durham, NC 27703
 E-Mail: cliffwangmail@yahoo.com"

DESCRIPTION

"This MIB module defines configuration objects for managing IPsec Security Policies. In general, this MIB can be implemented anywhere IPsec security services exist (e.g., bump-in-the-wire, host, gateway, firewall, router, etc.).

Copyright (C) The IETF Trust (2007). This version of this MIB module is part of RFC 4807; see the RFC itself for full legal notices."

-- Revision History

REVISION "200702070000Z" -- 7 February 2007
 DESCRIPTION "Initial version, published as RFC 4807."

::= { mib-2 153 }

--
 -- groups of related objects
 --

spdConfigObjects	OBJECT IDENTIFIER
::= { spdMIB 1 }	
spdNotificationObjects	OBJECT IDENTIFIER
::= { spdMIB 2 }	
spdConformanceObjects	OBJECT IDENTIFIER
::= { spdMIB 3 }	
spdActions	OBJECT IDENTIFIER
::= { spdMIB 4 }	

--
 -- Textual Conventions
 --

SpdBooleanOperator ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The SpdBooleanOperator operator is used to specify whether sub-components in a decision-making process are

ANDed or ORed together to decide if the resulting expression is true or false."

SYNTAX INTEGER { or(1), and(2) }

SpdAdminStatus ::= TEXTUAL-CONVENTION

STATUS current

DESCRIPTION

"The SpdAdminStatus is used to specify the administrative status of an object. Objects that are disabled MUST NOT be used by the packet processing engine."

SYNTAX INTEGER { enabled(1), disabled(2) }

SpdIPPacketLogging ::= TEXTUAL-CONVENTION

DISPLAY-HINT "d"

STATUS current

DESCRIPTION

"SpdIPPacketLogging specifies whether an audit message SHOULD be logged if a packet is passed through a Security Association (SA) and if some of that packet is included in the log event. A value of '-1' indicates no logging. A value of '0' or greater indicates that logging SHOULD be done and indicates the number of bytes starting at the beginning of the packet to place in the log. Values greater than the size of the packet being processed indicate that the entire packet SHOULD be sent.

Examples:

'-1' no logging

'0' log but do not include any of the packet in the log

'20' log and include the first 20 bytes of the packet in the log."

SYNTAX Integer32 (-1..65535)

SpdTimePeriod ::= TEXTUAL-CONVENTION

DISPLAY-HINT "31t"

STATUS current

DESCRIPTION

"This property identifies an overall range of calendar dates and time. In a boolean context, a value within this time range, inclusive, is considered true.

This information is encoded as an octet string using the UTF-8 transformation format described in STD 63, RFC 3629.

It uses the format suggested in RFC 3060. An octet string

represents a start date and time and an end date and time.
For example:

yyyymmddThhmmss/yyyymmddThhmmss

Where: yyyy = year mm = month dd = day
 hh = hour mm = minute ss = second

The first 'yyyymmddThhmmss' sub-string indicates the start date and time. The second 'yyyymmddThhmmss' sub-string indicates the end date and time. The character 'T' within these sub-strings indicates the beginning of the time portion of each sub-string. The solidus character '/' separates the start from the end date and time. The end date and time MUST be subsequent to the start date and time.

There are also two allowed substitutes for a 'yyyymmddThhmmss' sub-string: one for the start date and time, and one for the end date and time.

If the start date and time are replaced with the string 'THISANDPRIOR', this sub-string would indicate the current date and time and the previous dates and time.

If the end date and time are replaced with the string 'THISANDFUTURE', this sub-string would indicate the current date and time and the subsequent dates and time.

Any of the following SHOULD be considered a 'wrongValue' error:

- Setting a value with the end date and time earlier than or equal to the start date and time.
- Setting the start date and time to 'THISANDFUTURE'.
- Setting the end date and time to 'THISANDPRIOR'."

REFERENCE "RFC 3060, 3269"

SYNTAX OCTET STRING (SIZE (0..31))

--

-- Policy group definitions

--

spdLocalConfigObjects OBJECT IDENTIFIER

::= { spdConfigObjects 1 }

spdIngressPolicyGroupName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE(0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object indicates the global system policy group that is to be applied on ingress packets (i.e., arriving at an interface from a network) when a given endpoint does not contain a policy definition in the spdEndpointToGroupTable. Its value can be used as an index into the spdGroupContentsTable to retrieve a list of policies. A zero length string indicates that no system-wide policy exists and the default policy of 'drop' SHOULD be executed for ingress packets until one is imposed by either this object or by the endpoint processing a given packet.

This object MUST be persistent"

DEFVAL { "" }

::= { spdLocalConfigObjects 1 }

spdEgressPolicyGroupName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE(0..32))

MAX-ACCESS read-write

STATUS current

DESCRIPTION

"This object indicates the policy group containing the global system policy that is to be applied on egress packets (i.e., packets leaving an interface and entering a network) when a given endpoint does not contain a policy definition in the spdEndpointToGroupTable. Its value can be used as an index into the spdGroupContentsTable to retrieve a list of policies. A zero length string indicates that no system-wide policy exists and the default policy of 'drop' SHOULD be executed for egress packets until one is imposed by either this object or by the endpoint processing a given packet.

This object MUST be persistent"

DEFVAL { "" }

::= { spdLocalConfigObjects 2 }

spdEndpointToGroupTable OBJECT-TYPE

SYNTAX SEQUENCE OF SpdEndpointToGroupEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table maps policies (groupings) onto an endpoint (interface). A policy group assigned to an endpoint is then used to control access to the network traffic passing through that endpoint.

If an endpoint has been configured with a policy group and no rule within that policy group matches that packet, the default action in this case SHALL be to drop the packet.

If no policy group has been assigned to an endpoint, then the policy group specified by `spdIngressPolicyGroupName` MUST be used on traffic inbound from the network through that endpoint, and the policy group specified by `spdEgressPolicyGroupName` MUST be used for traffic outbound to the network through that endpoint."

```
::= { spdConfigObjects 2 }
```

`spdEndpointToGroupEntry` OBJECT-TYPE

SYNTAX `SpdEndpointToGroupEntry`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A mapping assigning a policy group to an endpoint."

INDEX { `spdEndGroupDirection`, `spdEndGroupInterface` }

```
::= { spdEndpointToGroupTable 1 }
```

`SpdEndpointToGroupEntry` ::= SEQUENCE {

`spdEndGroupDirection`

`IfDirection`,

`spdEndGroupInterface`

`InterfaceIndex`,

`spdEndGroupName`

`SnmpAdminString`,

`spdEndGroupLastChanged`

`TimeStamp`,

`spdEndGroupStorageType`

`StorageType`,

`spdEndGroupRowStatus`

`RowStatus`

}

`spdEndGroupDirection` OBJECT-TYPE

SYNTAX `IfDirection`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This object indicates which direction of packets crossing the interface are associated with which `spdEndGroupName` object. Ingress packets, or packets into the device match when this value is `inbound(1)`. Egress packets or packets out of the device match when this value is `outbound(2)`."

```
::= { spdEndpointToGroupEntry 1 }
```

`spdEndGroupInterface` OBJECT-TYPE

SYNTAX `InterfaceIndex`

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This value matches the IF-MIB's ifTable's ifIndex column and indicates the interface associated with a given endpoint. This object can be used to uniquely identify an endpoint that a set of policy groups are applied to."

::= { spdEndpointToGroupEntry 2 }

spdEndGroupName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE(1..32))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The policy group name to apply at this endpoint. The value of the spdEndGroupName object is then used as an index into the spdGroupContentsTable to come up with a list of rules that MUST be applied at this endpoint."

::= { spdEndpointToGroupEntry 3 }

spdEndGroupLastChanged OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdEndpointToGroupEntry 4 }

spdEndGroupStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }

::= { spdEndpointToGroupEntry 5 }

spdEndGroupRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

This object is considered 'notReady' and MUST NOT be set to active until one or more active rows exist within the spdGroupContentsTable for the group referenced by the spdEndGroupName object."

::= { spdEndpointToGroupEntry 6 }

--

-- policy group definition table

--

spdGroupContentsTable OBJECT-TYPE

SYNTAX SEQUENCE OF SpdGroupContentsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table contains a list of rules and/or subgroups contained within a given policy group. For a given value of spdGroupContName, the set of rows sharing that value forms a 'group'. The rows in a group MUST be processed according to the value of the spdGroupContPriority object in each row. The processing MUST be executed starting with the lowest value of spdGroupContPriority and in ascending order thereafter.

If an action is executed as the result of the processing of a row in a group, the processing of further rows in that group MUST stop. Iterating to the next policy group row by finding the next largest spdGroupContPriority object SHALL only be done if no actions were run while processing the current row for a given packet."

::= { spdConfigObjects 3 }

spdGroupContentsEntry OBJECT-TYPE

SYNTAX SpdGroupContentsEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"Defines a given sub-component within a policy group. A sub-component is either a rule or another group as indicated by spdGroupContComponentType and referenced by spdGroupContComponentName."


```

INDEX    { spdGroupContName, spdGroupContPriority }
 ::= { spdGroupContentsTable 1 }

```

```

SpdGroupContentsEntry ::= SEQUENCE {
    spdGroupContName          SnmpAdminString,
    spdGroupContPriority      Integer32,
    spdGroupContFilter        VariablePointer,
    spdGroupContComponentType INTEGER,
    spdGroupContComponentName SnmpAdminString,
    spdGroupContLastChanged   TimeStamp,
    spdGroupContStorageType   StorageType,
    spdGroupContRowStatus     RowStatus
}

```

```

spdGroupContName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The administrative name of the group associated with this
        row. A 'group' is formed by all the rows in this table that
        have the same value of this object."
    ::= { spdGroupContentsTable 1 }

```

```

spdGroupContPriority OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "The priority (sequence number) of the sub-component in
        a group that this row represents. This value indicates
        the order that each row of this table MUST be processed
        from low to high. For example, a row with a priority of 0
        is processed before a row with a priority of 1, a 1 before
        a 2, etc."
    ::= { spdGroupContentsTable 2 }

```

```

spdGroupContFilter OBJECT-TYPE
    SYNTAX      VariablePointer
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "spdGroupContFilter points to a filter that is evaluated
        to determine whether the spdGroupContComponentName within
        this row is exercised. Managers can use this object to
        classify groups of rules, or subgroups, together in order to
        achieve a greater degree of control and optimization over
        the execution order of the items within the group. If the

```

filter evaluates to false, the rule or subgroup will be skipped and the next rule or subgroup will be evaluated instead. This value can be used to indicate a scalar or row in a table. When indicating a row in a table, this value MUST point to the first column instance in that row.

An example usage of this object would be to limit a group of rules to executing only when the IP packet being processed is designated to be processed by IKE. This effectively creates a group of IKE-specific rules.

The following tables and scalars can be pointed to by this column. All but diffServMultiFieldClfrTable are defined in this MIB:

```
diffServMultiFieldClfrTable
spdIpOffsetFilterTable
spdTimeFilterTable
spdCompoundFilterTable
spdTrueFilter
spdIpsoHeaderFilterTable
```

Implementations MAY choose to provide support for other filter tables or scalars.

If this column is set to a VariablePointer value, which references a non-existent row in an otherwise supported table, the inconsistentName exception MUST be returned. If the table or scalar pointed to by the VariablePointer is not supported at all, then an inconsistentValue exception MUST be returned.

If, during packet processing, a row in this table is applied to a packet and the value of this column in that row references a non-existent or non-supported object, the packet MUST be dropped."

REFERENCE "RFC 3289"

```
DEFVAL { spdTrueFilterInstance }
::= { spdGroupContentsEntry 3 }
```

spdGroupContComponentType OBJECT-TYPE

```
SYNTAX      INTEGER { group(1), rule(2) }
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"Indicates whether the spdGroupContComponentName object is the name of another group defined within the spdGroupContentsTable or is the name of a rule defined

within the spdRuleDefinitionTable."
 DEFVAL { rule }
 ::= { spdGroupContentsEntry 4 }

spdGroupContComponentName OBJECT-TYPE
 SYNTAX SnmpAdminString (SIZE(1..32))
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The name of the policy rule or subgroup contained within
 this row, as indicated by the spdGroupContComponentType
 object."
 ::= { spdGroupContentsEntry 5 }

spdGroupContLastChanged OBJECT-TYPE
 SYNTAX TimeStamp
 MAX-ACCESS read-only
 STATUS current
 DESCRIPTION
 "The value of sysUpTime when this row was last modified
 or created either through SNMP SETs or by some other
 external means.

 If this row has not been modified since the last
 re-initialization of the network management subsystem,
 this object SHOULD have a zero value."
 ::= { spdGroupContentsEntry 6 }

spdGroupContStorageType OBJECT-TYPE
 SYNTAX StorageType
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "The storage type for this row. Rows in this table that
 were created through an external process MAY have a storage
 type of readOnly or permanent.

 For a storage type of permanent, none of the columns have
 to be writable."
 DEFVAL { nonVolatile }
 ::= { spdGroupContentsEntry 7 }

spdGroupContRowStatus OBJECT-TYPE
 SYNTAX RowStatus
 MAX-ACCESS read-create
 STATUS current
 DESCRIPTION
 "This object indicates the conceptual status of this row."

The value of this object has no effect on whether other objects in this conceptual row can be modified.

This object MUST NOT be set to active until the row to which the `spdGroupContComponentName` points to exists and is active.

If active, this object MUST remain active unless one of the following two conditions are met:

- I. No active row in `spdEndpointToGroupTable` exists that references this row's group (i.e., indicate this row's `spdGroupContName`).
- II. Or at least one other active row in this table has a matching `spdGroupContName`.

If neither condition is met, an attempt to set this row to something other than active MUST result in an `inconsistentValue` error."

```
::= { spdGroupContentsEntry 8 }
```

```
--
```

```
-- policy definition table
```

```
--
```

```
spdRuleDefinitionTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF SpdRuleDefinitionEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This table defines a rule by associating a filter
or a set of filters to an action to be executed."
```

```
::= { spdConfigObjects 4 }
```

```
spdRuleDefinitionEntry OBJECT-TYPE
```

```
SYNTAX      SpdRuleDefinitionEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"A row defining a particular rule definition. A rule
definition binds a filter pointer to an action pointer."
```

```
INDEX      { spdRuleDefName }
```

```
::= { spdRuleDefinitionTable 1 }
```

```
SpdRuleDefinitionEntry ::= SEQUENCE {
```

```
    spdRuleDefName
```

```
    SnmpAdminString,
```

```

    spdRuleDefDescription      SnmpAdminString,
    spdRuleDefFilter           VariablePointer,
    spdRuleDefFilterNegated    TruthValue,
    spdRuleDefAction           VariablePointer,
    spdRuleDefAdminStatus      SpdAdminStatus,
    spdRuleDefLastChanged      TimeStamp,
    spdRuleDefStorageType      StorageType,
    spdRuleDefRowStatus        RowStatus
}

```

```

spdRuleDefName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "spdRuleDefName is the administratively assigned name of
        the rule referred to by the spdGroupContComponentName
        object."
    ::= { spdRuleDefinitionEntry 1 }

```

```

spdRuleDefDescription OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "A user defined string. This field MAY be used for
        administrative tracking purposes."
    DEFVAL { "" }
    ::= { spdRuleDefinitionEntry 2 }

```

```

spdRuleDefFilter OBJECT-TYPE
    SYNTAX      VariablePointer
    MAX-ACCESS   read-create
    STATUS      current
    DESCRIPTION
        "spdRuleDefFilter points to a filter that is used to
        evaluate whether the action associated with this row is
        executed or not. The action will only execute if the
        filter referenced by this object evaluates to TRUE after
        first applying any negation required by the
        spdRuleDefFilterNegated object.

```

The following tables and scalars can be pointed to by this column. All but diffServMultiFieldClfrTable are defined in this MIB. Implementations MAY choose to provide support for other filter tables or scalars as well:

diffServMultiFieldClfrTable

```

spdIpOffsetFilterTable
spdTimeFilterTable
spdCompoundFilterTable
spdTrueFilter

```

If this column is set to a VariablePointer value, which references a non-existent row in an otherwise supported table, the inconsistentName exception MUST be returned. If the table or scalar pointed to by the VariablePointer is not supported at all, then an inconsistentValue exception MUST be returned.

If, during packet processing, this column has a value that references a non-existent or non-supported object, the packet MUST be dropped."

REFERENCE "RFC 3289"

```
 ::= { spdRuleDefinitionEntry 3 }
```

spdRuleDefFilterNegated OBJECT-TYPE

```

SYNTAX      TruthValue
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION

```

"spdRuleDefFilterNegated specifies whether or not the results of the filter referenced by the spdRuleDefFilter object is negated."

```
DEFVAL { false }
```

```
 ::= { spdRuleDefinitionEntry 4 }
```

spdRuleDefAction OBJECT-TYPE

```

SYNTAX      VariablePointer
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION

```

"This column points to the action to be taken. It MAY, but is not limited to, point to a row in one of the following tables:

```

    spdCompoundActionTable
    ipsaSaPreconfiguredActionTable
    ipiaIkeActionTable
    ipiaIpsecActionTable

```

It MAY also point to one of the scalar objects beneath spdStaticActions.

If this object is set to a pointer to a row in an unsupported (or unknown) table, an inconsistentValue

error MUST be returned.

If this object is set to point to a non-existent row in an otherwise supported table, an inconsistentName error MUST be returned.

If, during packet processing, this column has a value that references a non-existent or non-supported object, the packet MUST be dropped."

::= { spdRuleDefinitionEntry 5 }

spdRuleDefAdminStatus OBJECT-TYPE

SYNTAX SpdAdminStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates whether the current rule definition is considered active. If the value is enabled, the rule MUST be evaluated when processing packets. If the value is disabled, the packet processing MUST continue as if this rule's filter had effectively failed."

DEFVAL { enabled }

::= { spdRuleDefinitionEntry 6 }

spdRuleDefLastChanged OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdRuleDefinitionEntry 7 }

spdRuleDefStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have

```

        to be writable."
DEFVAL { nonVolatile }
 ::= { spdRuleDefinitionEntry 8 }

```

spdRuleDefRowStatus OBJECT-TYPE

```

SYNTAX      RowStatus
MAX-ACCESS  read-create
STATUS      current
DESCRIPTION

```

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

This object MUST NOT be set to active until the containing conditions, filters, and actions have been defined. Once active, it MUST remain active until no active policyGroupContents entries are referencing it. A failed attempt to do so MUST return an inconsistentValue error."

```

 ::= { spdRuleDefinitionEntry 9 }

```

```

--
-- Policy compound filter definition table
--

```

spdCompoundFilterTable OBJECT-TYPE

```

SYNTAX      SEQUENCE OF SpdCompoundFilterEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"A table defining compound filters and their associated parameters. A row in this table can be pointed to by a spdRuleDefFilter object."

```

 ::= { spdConfigObjects 5 }

```

spdCompoundFilterEntry OBJECT-TYPE

```

SYNTAX      SpdCompoundFilterEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION

```

"An entry in the spdCompoundFilterTable. Each entry in this table represents a compound filter. A filter defined by this table is considered to have a TRUE return value if and only if:

spdCompFiltLogicType is AND and all of the sub-filters associated with it, as defined in the spdSubfiltersTable, are all true themselves (after applying any required

negation, as defined by the ficFilterIsNegated object).

spdCompFiltLogicType is OR and at least one of the sub-filters associated with it, as defined in the spdSubfiltersTable, is true itself (after applying any required negation, as defined by the ficFilterIsNegated object."

```
INDEX      { spdCompFiltName }
 ::= { spdCompoundFilterTable 1 }
```

```
SpdCompoundFilterEntry ::= SEQUENCE {
    spdCompFiltName                SnmpAdminString,
    spdCompFiltDescription         SnmpAdminString,
    spdCompFiltLogicType           SpdBooleanOperator,
    spdCompFiltLastChanged         TimeStamp,
    spdCompFiltStorageType         StorageType,
    spdCompFiltRowStatus           RowStatus
}
```

```
spdCompFiltName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A user definable string. This value is used as an index
         into this table."
    ::= { spdCompoundFilterEntry 1 }
```

```
spdCompFiltDescription OBJECT-TYPE
    SYNTAX      SnmpAdminString
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "A user definable string. This field MAY be used for
         your administrative tracking purposes."
    DEFVAL { "" }
    ::= { spdCompoundFilterEntry 2 }
```

```
spdCompFiltLogicType OBJECT-TYPE
    SYNTAX      SpdBooleanOperator
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "Indicates whether the sub-component filters of this
         compound filter are functionally ANDed or ORed together."
    DEFVAL { and }
    ::= { spdCompoundFilterEntry 3 }
```

spdCompFiltLastChanged OBJECT-TYPE

SYNTAX TimeStamp
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdCompoundFilterEntry 4 }

spdCompFiltStorageType OBJECT-TYPE

SYNTAX StorageType
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }

::= { spdCompoundFilterEntry 5 }

spdCompFiltRowStatus OBJECT-TYPE

SYNTAX RowStatus
 MAX-ACCESS read-create
 STATUS current

DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

Once active, it MUST NOT have its value changed if any active rows in the spdRuleDefinitionTable are currently pointing at this row."

::= { spdCompoundFilterEntry 6 }

--

-- Policy filters in a cf table

--

spdSubfiltersTable OBJECT-TYPE

SYNTAX SEQUENCE OF SpdSubfiltersEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"This table defines a list of filters contained within a given compound filter defined in the spdCompoundFilterTable."

::= { spdConfigObjects 6 }

spdSubfiltersEntry OBJECT-TYPE

SYNTAX SpdSubfiltersEntry

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"An entry in the spdSubfiltersTable. There is an entry in this table for each sub-filter of all compound filters present in the spdCompoundFilterTable."

INDEX { spdCompFiltName, spdSubFiltPriority }

::= { spdSubfiltersTable 1 }

SpdSubfiltersEntry ::= SEQUENCE {

spdSubFiltPriority

Integer32,

spdSubFiltSubfilter

VariablePointer,

spdSubFiltSubfilterIsNegated

TruthValue,

spdSubFiltLastChanged

TimeStamp,

spdSubFiltStorageType

StorageType,

spdSubFiltRowStatus

RowStatus

}

spdSubFiltPriority OBJECT-TYPE

SYNTAX Integer32 (0..65535)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"The priority of a given filter within a compound filter. The order of execution is from lowest to highest priority value (i.e., priority 0 before priority 1, 1 before 2, etc.). Implementations MAY choose to follow this ordering, as set by the manager that created the rows. This can allow a manager to intelligently construct filter lists such that faster filters are evaluated first."

::= { spdSubfiltersEntry 1 }

spdSubFiltSubfilter OBJECT-TYPE

SYNTAX VariablePointer

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The OID of the contained filter. The value of this object is a VariablePointer that references the filter to be included in this compound filter.

The following tables and scalars can be pointed to by this column. All but diffServMultiFieldClfrTable are defined in this MIB. Implementations MAY choose to provide support for other filter tables or scalars as well:

```
diffServMultiFieldClfrTable
spdIpsoHeaderFilterTable
spdIpOffsetFilterTable
spdTimeFilterTable
spdCompoundFilterTable
spdTrueFilter
```

If this column is set to a VariablePointer value that references a non-existent row in an otherwise supported table, the inconsistentName exception MUST be returned. If the table or scalar pointed to by the VariablePointer is not supported at all, then an inconsistentValue exception MUST be returned.

If, during packet processing, this column has a value that references a non-existent or non-supported object, the packet MUST be dropped."

REFERENCE "RFC 3289"

```
::= { spdSubfiltersEntry 2 }
```

spdSubFiltSubfilterIsNegated OBJECT-TYPE

SYNTAX TruthValue

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"Indicates whether or not the result of applying this sub-filter is negated."

DEFVAL { false }

```
::= { spdSubfiltersEntry 3 }
```

spdSubFiltLastChanged OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."
 ::= { spdSubfiltersEntry 4 }

spdSubFiltStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }
 ::= { spdSubfiltersEntry 5 }

spdSubFiltRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

This object cannot be made active until a filter referenced by the spdSubFiltSubfilter object is both defined and active. An attempt to do so MUST result in an inconsistentValue error.

If active, this object MUST remain active unless one of the following two conditions are met:

- I. No active row in the SpdCompoundFilterTable exists that has a matching spdCompFiltName.
- II. Or, at least one other active row in this table has a matching spdCompFiltName.

If neither condition is met, an attempt to set this row to something other than active MUST result in an inconsistentValue error."

::= { spdSubfiltersEntry 6 }

```
--
-- Static Filters
--

spdStaticFilters OBJECT IDENTIFIER ::= { spdConfigObjects 7 }

spdTrueFilter OBJECT-TYPE
    SYNTAX      Integer32 (1)
    MAX-ACCESS   read-only
    STATUS      current
    DESCRIPTION
        "This scalar indicates a (automatic) true result for
        a filter. That is, this is a filter that is always
        true; it is useful for adding as a default filter for a
        default action or a set of actions."
    ::= { spdStaticFilters 1 }

spdTrueFilterInstance OBJECT IDENTIFIER ::= { spdTrueFilter 0 }

--
-- Policy IP Offset filter definition table
--

spdIpOffsetFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SpdIpOffsetFilterEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains a list of filter definitions to be
        used within the spdRuleDefinitionTable or the
        spdSubfiltersTable.

        This type of filter is used to compare an administrator
        specified octet string to the octets at a particular
        location in a packet."
    ::= { spdConfigObjects 8 }

spdIpOffsetFilterEntry OBJECT-TYPE
    SYNTAX      SpdIpOffsetFilterEntry
    MAX-ACCESS   not-accessible
    STATUS      current
    DESCRIPTION
        "A definition of a particular filter."
    INDEX       { spdIpOffsetFilterName }
    ::= { spdIpOffsetFilterTable 1 }
```

```

SpdIpOffsetFilterEntry ::= SEQUENCE {
    spdIpOffFiltName          SnmpAdminString,
    spdIpOffFiltOffset        Unsigned32,
    spdIpOffFiltType          INTEGER,
    spdIpOffFiltValue         OCTET STRING,
    spdIpOffFiltLastChanged   TimeStamp,
    spdIpOffFiltStorageType   StorageType,
    spdIpOffFiltRowStatus     RowStatus
}

spdIpOffFiltName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The administrative name for this filter."
    ::= { spdIpOffsetFilterEntry 1 }

spdIpOffFiltOffset OBJECT-TYPE
    SYNTAX      Unsigned32 (0..65535)
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This is the byte offset from the front of the entire IP
        packet where the value or arithmetic comparison is done.  A
        value of '0' indicates the first byte of the packet header.
        If this value is greater than the length of the packet, the
        filter represented by this row should be considered to
        fail."
    ::= { spdIpOffsetFilterEntry 2 }

spdIpOffFiltType OBJECT-TYPE
    SYNTAX INTEGER { equal(1),
                    notEqual(2),
                    arithmeticLess(3),
                    arithmeticGreaterOrEqual(4),
                    arithmeticGreater(5),
                    arithmeticLessOrEqual(6) }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This defines the various tests that are used when
        evaluating a given filter.

        The various tests definable in this table are as follows:

        equal:
            - Tests if the OCTET STRING, 'spdIpOffFiltValue', matches

```

a value in the packet starting at the given offset in the packet and comparing the entire OCTET STRING of 'spdIpOffFiltValue'. Any values compared this way are assumed to be unsigned integer values in network byte order of the same length as 'spdIpOffFiltValue'.

notEqual:

- Tests if the OCTET STRING, 'spdIpOffFiltValue', does not match a value in the packet starting at the given offset in the packet and comparing to the entire OCTET STRING of 'spdIpOffFiltValue'. Any values compared this way are assumed to be unsigned integer values in network byte order of the same length as 'spdIpOffFiltValue'.

arithmeticLess:

- Tests if the OCTET STRING, 'spdIpOffFiltValue', is arithmetically less than ('<') the value starting at the given offset within the packet. The value in the packet is assumed to be an unsigned integer in network byte order of the same length as 'spdIpOffFiltValue'.

arithmeticGreaterOrEqual:

- Tests if the OCTET STRING, 'spdIpOffFiltValue', is arithmetically greater than or equal to ('>=') the value starting at the given offset within the packet. The value in the packet is assumed to be an unsigned integer in network byte order of the same length as 'spdIpOffFiltValue'.

arithmeticGreater:

- Tests if the OCTET STRING, 'spdIpOffFiltValue', is arithmetically greater than ('>') the value starting at the given offset within the packet. The value in the packet is assumed to be an unsigned integer in network byte order of the same length as 'spdIpOffFiltValue'.

arithmeticLessOrEqual:

- Tests if the OCTET STRING, 'spdIpOffFiltValue', is arithmetically less than or equal to ('<=') the value starting at the given offset within the packet. The value in the packet is assumed to be an unsigned integer in network byte order of the same length as 'spdIpOffFiltValue'."

::= { spdIpOffsetFilterEntry 3 }

spdIpOffFiltValue OBJECT-TYPE

SYNTAX OCTET STRING (SIZE(1..1024))

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"spdIpOfffiltValue is used for match comparisons of a packet at spdIpOfffiltOffset."

::= { spdIpOffsetFilterEntry 4 }

spdIpOfffiltLastChanged OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdIpOffsetFilterEntry 5 }

spdIpOfffiltStorageType OBJECT-TYPE

SYNTAX StorageType

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }

::= { spdIpOffsetFilterEntry 6 }

spdIpOfffiltRowStatus OBJECT-TYPE

SYNTAX RowStatus

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

If active, this object MUST remain active if it is

referenced by an active row in another table. An attempt to set it to anything other than active while it is referenced by an active row in another table MUST result in an inconsistentValue error."

```
::= { spdIpOffsetFilterEntry 7 }
```

```
--
```

```
-- Time/scheduling filter table
```

```
--
```

```
spdTimeFilterTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF SpdTimeFilterEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

"Defines a table of filters that can be used to effectively enable or disable policies based on a valid time range."

```
::= { spdConfigObjects 9 }
```

```
spdTimeFilterEntry OBJECT-TYPE
```

```
SYNTAX      SpdTimeFilterEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

"A row describing a given time frame for which a policy is filtered on to activate or deactivate the rule."

If all the column objects in a row are true for the current time, the row evaluates as 'true'. More explicitly, the time matching column objects in a row MUST be logically ANDed together to form the boolean true/false for the row."

```
INDEX      { spdTimeFiltName }
```

```
::= { spdTimeFilterTable 1 }
```

```
SpdTimeFilterEntry ::= SEQUENCE {
```

```
    spdTimeFiltName          SnmpAdminString,
```

```
    spdTimeFiltPeriod        SpdTimePeriod,
```

```
    spdTimeFiltMonthOfYearMask BITS,
```

```
    spdTimeFiltDayOfMonthMask OCTET STRING,
```

```
    spdTimeFiltDayOfWeekMask BITS,
```

```
    spdTimeFiltTimeOfDayMask SpdTimePeriod,
```

```
    spdTimeFiltLastChanged   TimeStamp,
```

```
    spdTimeFiltStorageType   StorageType,
```

```
    spdTimeFiltRowStatus     RowStatus
```

```
}
```

```

spdTimeFiltName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION
        "An administratively assigned name for this filter."
    ::= { spdTimeFilterEntry 1 }

spdTimeFiltPeriod OBJECT-TYPE
    SYNTAX      SpdTimePeriod
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "The valid time period for this filter. This column is
        considered 'true' if the current time is within the range of
        this object."
    DEFVAL { "THISANDPRIOR/THISANDFUTURE" }
    ::= { spdTimeFilterEntry 2 }

spdTimeFiltMonthOfYearMask OBJECT-TYPE
    SYNTAX      BITS { january(0), february(1), march(2),
                        april(3), may(4), june(5), july(6),
                        august(7), september(8), october(9),
                        november(10), december(11) }
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "A bit mask that indicates acceptable months of the year.
        This column evaluates to 'true' if the current month's bit
        is set."
    DEFVAL { { january, february, march, april, may, june, july,
                august, september, october, november, december } }
    ::= { spdTimeFilterEntry 3 }

spdTimeFiltDayOfMonthMask OBJECT-TYPE
    SYNTAX      OCTET STRING (SIZE(8))
    MAX-ACCESS   read-create
    STATUS       current
    DESCRIPTION
        "Defines which days of the month the current time is
        valid for. It is a sequence of 64 BITS, where each BIT
        represents a corresponding day of the month in forward or
        reverse order. Starting from the left-most bit, the first
        31 bits identify the day of the month, counting from the
        beginning of the month. The following 31 bits (bits 32-62)
        indicate the day of the month, counting from the end of the

```

month. For months with fewer than 31 days, the bits that correspond to the non-existent days of that month are ignored (e.g., for non-leap year Februarys, bits 29-31 and 60-62 are ignored).

This column evaluates to 'true' if the current day of the month's bit is set.

For example, a value of 0x'80 00 00 01 00 00 00 00' indicates that this column evaluates to true on the first and last days of the month.

The last two bits in the string MUST be zero."

```
DEFVAL { 'fffffffffffffe'H }
::= { spdTimeFilterEntry 4 }
```

spdTimeFiltDayOfWeekMask OBJECT-TYPE

```
SYNTAX      BITS { sunday(0), monday(1), tuesday(2),
                  wednesday(3), thursday(4), friday(5),
                  saturday(6) }
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"A bit mask that defines which days of the week that the current time is valid for. This column evaluates to 'true' if the current day of the week's bit is set."

```
DEFVAL { { monday, tuesday, wednesday, thursday, friday,
          saturday, sunday } }
::= { spdTimeFilterEntry 5 }
```

spdTimeFiltTimeOfDayMask OBJECT-TYPE

```
SYNTAX      SpdTimePeriod
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

DESCRIPTION

"Indicates the start and end time of the day for which this filter evaluates to true. The date portions of the spdTimePeriod TC are ignored for purposes of evaluating this mask, and only the time-specific portions are used.

This column evaluates to 'true' if the current time of day is within the range of the start and end times of the day indicated by this object."

```
DEFVAL { "00000000T000000/00000000T240000" }
::= { spdTimeFilterEntry 6 }
```

spdTimeFiltLastChanged OBJECT-TYPE

```
SYNTAX      TimeStamp
```

MAX-ACCESS read-only
STATUS current
DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdTimeFilterEntry 7 }

spdTimeFiltStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }

::= { spdTimeFilterEntry 8 }

spdTimeFiltRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current
DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

If active, this object MUST remain active if it is referenced by an active row in another table. An attempt to set it to anything other than active while it is referenced by an active row in another table MUST result in an inconsistentValue error."

::= { spdTimeFilterEntry 9 }

--

-- IPSO protection authority filtering

--

```

spdIpsoHeaderFilterTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF SpdIpsoHeaderFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This table contains a list of IPSO header filter
        definitions to be used within the spdRuleDefinitionTable or
        the spdSubfiltersTable.  IPSO headers and their values are
        described in RFC 1108."
    REFERENCE   "RFC 1108"
    ::= { spdConfigObjects 10 }

```

```

spdIpsoHeaderFilterEntry OBJECT-TYPE
    SYNTAX      SpdIpsoHeaderFilterEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A definition of a particular filter."
    INDEX       { spdIpsoHeadFiltName }
    ::= { spdIpsoHeaderFilterTable 1 }

```

```

SpdIpsoHeaderFilterEntry ::= SEQUENCE {
    spdIpsoHeadFiltName          SnmpAdminString,
    spdIpsoHeadFiltType          BITS,
    spdIpsoHeadFiltClassification  INTEGER,
    spdIpsoHeadFiltProtectionAuth  INTEGER,
    spdIpsoHeadFiltLastChanged    TimeStamp,
    spdIpsoHeadFiltStorageType    StorageType,
    spdIpsoHeadFiltRowStatus      RowStatus
}

```

```

spdIpsoHeadFiltName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The administrative name for this filter."
    ::= { spdIpsoHeaderFilterEntry 1 }

```

```

spdIpsoHeadFiltType OBJECT-TYPE
    SYNTAX      BITS { classificationLevel(0),
                      protectionAuthority(1) }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates which of the IPSO header field a
        packet is filtered on for this row.  If this object is set
        to classification(0), the spdIpsoHeadFiltClassification

```

object indicates how the packet is filtered. If this object is set to protectionAuthority(1), the spdIpsoHeadFiltProtectionAuth object indicates how the packet is filtered."

::= { spdIpsoHeaderFilterEntry 2 }

spdIpsoHeadFiltClassification OBJECT-TYPE

SYNTAX INTEGER { topSecret(61), secret(90),
confidential(150), unclassified(171) }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicates the IPSO classification header field value that the packet MUST have for this row to evaluate to 'true'.

The values of these enumerations are defined by RFC 1108."

REFERENCE "RFC 1108"

::= { spdIpsoHeaderFilterEntry 3 }

spdIpsoHeadFiltProtectionAuth OBJECT-TYPE

SYNTAX INTEGER { genser(0), siopesi(1), sci(2),
nsa(3), doe(4) }

MAX-ACCESS read-create

STATUS current

DESCRIPTION

"This object indicates the IPSO protection authority header field value that the packet MUST have for this row to evaluate to 'true'.

The values of these enumerations are defined by RFC 1108.

Hence the reason the SMIV2 convention of not using 0 in enumerated lists is violated here."

REFERENCE "RFC 1108"

::= { spdIpsoHeaderFilterEntry 4 }

spdIpsoHeadFiltLastChanged OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

```
::= { spdIpsoHeaderFilterEntry 5 }
```

```
spdIpsoHeadFiltStorageType OBJECT-TYPE
```

```
SYNTAX      StorageType
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

```
DEFVAL { nonVolatile }
```

```
::= { spdIpsoHeaderFilterEntry 6 }
```

```
spdIpsoHeadFiltRowStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

However, this object MUST NOT be set to active if the requirements of the spdIpsoHeadFiltType object are not met. Specifically, if the spdIpsoHeadFiltType bit for classification(0) is set, the spdIpsoHeadFiltClassification column MUST have a valid value for the row status to be set to active. If the spdIpsoHeadFiltType bit for protectionAuthority(1) is set, the spdIpsoHeadFiltProtectionAuth column MUST have a valid value for the row status to be set to active.

If active, this object MUST remain active if it is referenced by an active row in another table. An attempt to set it to anything other than active while it is referenced by an active row in another table MUST result in an inconsistentValue error."

```
::= { spdIpsoHeaderFilterEntry 7 }
```

```
--
```

```
-- compound actions table
```

```
--
```

```
spdCompoundActionTable OBJECT-TYPE
```



```

SYNTAX      SEQUENCE OF SpdCompoundActionEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Table used to allow multiple actions to be associated
    with a rule.  It uses the spdSubactionsTable to do this.
    The rows from spdSubactionsTable that are partially indexed
    by spdCompActName form the set of compound actions to be
    performed.  The spdCompActExecutionStrategy column in this
    table indicates how those actions are processed."
 ::= { spdConfigObjects 11 }

```

```

spdCompoundActionEntry OBJECT-TYPE
    SYNTAX      SpdCompoundActionEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "A row in the spdCompoundActionTable."
    INDEX       { spdCompActName }
    ::= { spdCompoundActionTable 1 }

```

```

SpdCompoundActionEntry ::= SEQUENCE {
    spdCompActName                SnmpAdminString,
    spdCompActExecutionStrategy   INTEGER,
    spdCompActLastChanged         TimeStamp,
    spdCompActStorageType         StorageType,
    spdCompActRowStatus           RowStatus
}

```

```

spdCompActName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "This is an administratively assigned name of this
        compound action."
    ::= { spdCompoundActionEntry 1 }

```

```

spdCompActExecutionStrategy OBJECT-TYPE
    SYNTAX      INTEGER { doAll(1),
                          doUntilSuccess(2),
                          doUntilFailure(3) }
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This object indicates how the sub-actions are executed
        based on the success of the actions as they finish
        executing."

```

- doAll - run each sub-action regardless of the exit status of the previous action. This parent action is always considered to have acted successfully.
- doUntilSuccess - run each sub-action until one succeeds, at which point stop processing the sub-actions within this parent compound action. If one of the sub-actions did execute successfully, this parent action is also considered to have executed successfully.
- doUntilFailure - run each sub-action until one fails, at which point stop processing the sub-actions within this compound action. If any sub-action fails, the result of this parent action is considered to have failed."

```
DEFVAL { doUntilSuccess }
::= { spdCompoundActionEntry 2 }
```

spdCompActLastChanged OBJECT-TYPE

```
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
```

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

```
::= { spdCompoundActionEntry 3 }
```

spdCompActStorageType OBJECT-TYPE

```
SYNTAX      StorageType
MAX-ACCESS  read-create
STATUS      current
```

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

```
DEFVAL { nonVolatile }
```

```
::= { spdCompoundActionEntry 4 }
```

```
spdCompActRowStatus OBJECT-TYPE
```

```
SYNTAX      RowStatus
```

```
MAX-ACCESS  read-create
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This object indicates the conceptual status of this row.
```

```
The value of this object has no effect on whether other
objects in this conceptual row can be modified.
```

```
Once a row in the spdCompoundActionTable has been made
active, this object MUST NOT be set to destroy without
first destroying all the contained rows listed in the
spdSubactionsTable."
```

```
::= { spdCompoundActionEntry 5 }
```

```
--
```

```
-- actions contained within a compound action
```

```
--
```

```
spdSubactionsTable OBJECT-TYPE
```

```
SYNTAX      SEQUENCE OF SpdSubactionsEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"This table contains a list of the sub-actions within a
given compound action. Compound actions executing these
actions MUST execute them in series based on the
spdSubActPriority value, with the lowest value executing
first."
```

```
::= { spdConfigObjects 12 }
```

```
spdSubactionsEntry OBJECT-TYPE
```

```
SYNTAX      SpdSubactionsEntry
```

```
MAX-ACCESS  not-accessible
```

```
STATUS      current
```

```
DESCRIPTION
```

```
"A row containing a reference to a given compound-action
sub-action."
```

```
INDEX      { spdCompActName, spdSubActPriority }
```

```
::= { spdSubactionsTable 1 }
```

```
SpdSubactionsEntry ::= SEQUENCE {
```

```
    spdSubActPriority
```

```
    Integer32,
```

```
    spdSubActSubActionName
```

```
    VariablePointer,
```

```

    spdSubActLastChanged      TimeStamp,
    spdSubActStorageType     StorageType,
    spdSubActRowStatus       RowStatus
}

spdSubActPriority OBJECT-TYPE
    SYNTAX      Integer32 (0..65535)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
        "The priority of a given sub-action within a compound
        action.  The order in which sub-actions MUST be executed
        are based on the value from this column, with the lowest
        numeric value executing first (i.e., priority 0 before
        priority 1, 1 before 2, etc.)."
    ::= { spdSubactionsEntry 1 }

spdSubActSubActionName OBJECT-TYPE
    SYNTAX      VariablePointer
    MAX-ACCESS  read-create
    STATUS      current
    DESCRIPTION
        "This column points to the action to be taken.  It MAY,
        but is not limited to, point to a row in one of the
        following tables:

            spdCompoundActionTable      - Allowing recursion
            ipsaSaPreconfiguredActionTable
            ipiaIkeActionTable
            ipiaIpsecActionTable

        It MAY also point to one of the scalar objects beneath
        spdStaticActions.

        If this object is set to a pointer to a row in an
        unsupported (or unknown) table, an inconsistentValue
        error MUST be returned.

        If this object is set to point to a non-existent row in
        an otherwise supported table, an inconsistentName error
        MUST be returned.

        If, during packet processing, this column has a value that
        references a non-existent or non-supported object, the
        packet MUST be dropped."
    ::= { spdSubactionsEntry 2 }

spdSubActLastChanged OBJECT-TYPE

```

SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current

DESCRIPTION

"The value of sysUpTime when this row was last modified or created either through SNMP SETs or by some other external means.

If this row has not been modified since the last re-initialization of the network management subsystem, this object SHOULD have a zero value."

::= { spdSubactionsEntry 3 }

spdSubActStorageType OBJECT-TYPE

SYNTAX StorageType
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"The storage type for this row. Rows in this table that were created through an external process MAY have a storage type of readOnly or permanent.

For a storage type of permanent, none of the columns have to be writable."

DEFVAL { nonVolatile }

::= { spdSubactionsEntry 4 }

spdSubActRowStatus OBJECT-TYPE

SYNTAX RowStatus
MAX-ACCESS read-create
STATUS current

DESCRIPTION

"This object indicates the conceptual status of this row.

The value of this object has no effect on whether other objects in this conceptual row can be modified.

If active, this object MUST remain active unless one of the following two conditions are met. An attempt to set it to anything other than active while the following conditions are not met MUST result in an inconsistentValue error. The two conditions are:

- I. No active row in the spdCompoundActionTable exists which has a matching spdCompActName.
- II. Or, at least one other active row in this table has a matching spdCompActName."

```
 ::= { spdSubactionsEntry 5 }

--
-- Static Actions
--

-- these are static actions that can be pointed to by the
-- spdRuleDefAction or the spdSubActSubActionName objects to
-- drop, accept, or reject packets.

spdStaticActions OBJECT IDENTIFIER ::= { spdConfigObjects 13 }

spdDropAction OBJECT-TYPE
    SYNTAX      Integer32 (1)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This scalar indicates that a packet MUST be dropped
         and SHOULD NOT have action/packet logging."
    ::= { spdStaticActions 1 }

spdDropActionLog OBJECT-TYPE
    SYNTAX      Integer32 (1)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This scalar indicates that a packet MUST be dropped
         and SHOULD have action/packet logging."
    ::= { spdStaticActions 2 }

spdAcceptAction OBJECT-TYPE
    SYNTAX      Integer32 (1)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This Scalar indicates that a packet MUST be accepted
         (pass-through) and SHOULD NOT have action/packet logging."
    ::= { spdStaticActions 3 }

spdAcceptActionLog OBJECT-TYPE
    SYNTAX      Integer32 (1)
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "This scalar indicates that a packet MUST be accepted
         (pass-through) and SHOULD have action/packet logging."
    ::= { spdStaticActions 4 }
```

```
--
--
-- Notification objects information
--
--

spdNotificationVariables OBJECT IDENTIFIER ::=
    { spdNotificationObjects 1 }

spdNotifications OBJECT IDENTIFIER ::=
    { spdNotificationObjects 0 }

spdActionExecuted OBJECT-TYPE
    SYNTAX      VariablePointer
    MAX-ACCESS   accessible-for-notify
    STATUS       current
    DESCRIPTION
        "Points to the action instance that was executed that
         resulted in the notification being sent."
    ::= { spdNotificationVariables 1 }

spdIPEndpointAddType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   accessible-for-notify
    STATUS       current
    DESCRIPTION
        "Contains the address type for the interface that the
         notification triggering packet is passing through."
    ::= { spdNotificationVariables 2 }

spdIPEndpointAddress OBJECT-TYPE
    SYNTAX      InetAddress
    MAX-ACCESS   accessible-for-notify
    STATUS       current
    DESCRIPTION
        "Contains the interface address for the interface that the
         notification triggering packet is passing through.

         The format of this object is specified by the
         spdIPEndpointAddType object."
    ::= { spdNotificationVariables 3 }

spdIPSourceType OBJECT-TYPE
    SYNTAX      InetAddressType
    MAX-ACCESS   accessible-for-notify
    STATUS       current
    DESCRIPTION
        "Contains the source address type of the packet that
```

triggered the notification."
::= { spdNotificationVariables 4 }

spdIPSourceAddress OBJECT-TYPE

SYNTAX InetAddress
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"Contains the source address of the packet that
triggered the notification.

The format of this object is specified by the
spdIPSourceType object."
::= { spdNotificationVariables 5 }

spdIPDestinationType OBJECT-TYPE

SYNTAX InetAddressType
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"Contains the destination address type of the packet
that triggered the notification."
::= { spdNotificationVariables 6 }

spdIPDestinationAddress OBJECT-TYPE

SYNTAX InetAddress
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"Contains the destination address of the packet that
triggered the notification.

The format of this object is specified by the
spdIPDestinationType object."
::= { spdNotificationVariables 7 }

spdPacketDirection OBJECT-TYPE

SYNTAX IfDirection
MAX-ACCESS accessible-for-notify
STATUS current
DESCRIPTION
"Indicates if the packet that triggered the action in
questions was ingress (inbound) or egress (outbound)."
::= { spdNotificationVariables 8 }

spdPacketPart OBJECT-TYPE

SYNTAX OCTET STRING (SIZE (0..65535))
MAX-ACCESS accessible-for-notify

STATUS current

DESCRIPTION

"spdPacketPart is the front part of the full IP packet that triggered this notification. The initial size limit is determined by the smaller of the size, indicated by:

- I. The value of the object with the TC syntax 'SpdIPPacketLogging' that indicated the packet SHOULD be logged and
- II. The size of the triggering packet.

The final limit is determined by the SNMP packet size when sending the notification. The maximum size that can be included will be the smaller of the initial size, given the above, and the length that will fit in a single SNMP notification packet after the rest of the notification's objects and any other necessary packet data (headers encoding, etc.) have been included in the packet."

::= { spdNotificationVariables 9 }

spdActionNotification NOTIFICATION-TYPE

OBJECTS { spdActionExecuted, spdIPEndpointAddType, spdIPEndpointAddress, spdIPSourceType, spdIPSourceAddress, spdIPDestinationType, spdIPDestinationAddress, spdPacketDirection }

STATUS current

DESCRIPTION

"Notification that an action was executed by a rule. Only actions with logging enabled will result in this notification getting sent. The object includes the spdActionExecuted object, which will indicate which action was executed within the scope of the rule. Additionally, the spdIPSourceType, spdIPSourceAddress, spdIPDestinationType, and spdIPDestinationAddress objects are included to indicate the packet source and destination of the packet that triggered the action. Finally, the spdIPEndpointAddType, spdIPEndpointAddress, and spdPacketDirection objects indicate which interface the executed action was associated with, and if the packet was ingress or egress through the endpoint.

A spdActionNotification SHOULD be limited to a maximum of one notification sent per minute for any action notifications that do not have any other configuration controlling their send rate.

Note that compound actions with multiple executed sub-actions may result in multiple notifications being sent from a single rule execution."

::= { spdNotifications 1 }

spdPacketNotification NOTIFICATION-TYPE

OBJECTS { spdActionExecuted, spdIPEndpointAddType,
spdIPEndpointAddress,
spdIPSourceType, spdIPSourceAddress,
spdIPDestinationType,
spdIPDestinationAddress,
spdPacketDirection,
spdPacketPart }

STATUS current

DESCRIPTION

"Notification that a packet passed through a Security Association (SA). Only SAs created by actions with packet logging enabled will result in this notification getting sent. The objects sent MUST include the spdActionExecuted, which will indicate which action was executed within the scope of the rule. Additionally, the spdIPSourceType, spdIPSourceAddress, spdIPDestinationType, and spdIPDestinationAddress objects MUST be included to indicate the packet source and destination of the packet that triggered the action. The spdIPEndpointAddType, spdIPEndpointAddress, and spdPacketDirection objects are included to indicate which endpoint the packet was associated with. Finally, spdPacketPart is included to enable sending a variable sized part of the front of the packet with the size dependent on the value of the object of TC syntax 'SpdIPPacketLogging', which indicated that logging should be done.

A spdPacketNotification SHOULD be limited to a maximum of one notification sent per minute for any action notifications that do not have any other configuration controlling their send rate.

An action notification SHOULD be limited to a maximum of one notification sent per minute for any action notifications that do not have any other configuration controlling their send rate."

::= { spdNotifications 2 }

--

--

-- Conformance information

```
--
--

spdCompliances OBJECT IDENTIFIER
    ::= { spdConformanceObjects 1 }
spdGroups OBJECT IDENTIFIER
    ::= { spdConformanceObjects 2 }

--
-- Compliance statements
--
--
spdRuleFilterFullCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for SNMP entities that include
        an IPsec MIB implementation with Endpoint, Rules, and
        filters support.

        When this MIB is implemented with support for read-create,
        then such an implementation can claim full compliance.  Such
        devices can then be both monitored and configured with this
        MIB."

MODULE -- This Module
    MANDATORY-GROUPS { spdEndpointGroup,
                        spdGroupContentsGroup,
                        spdRuleDefinitionGroup,
                        spdStaticFilterGroup,
                        spdStaticActionGroup ,
                        diffServMIBMultiFieldClfrGroup }

    GROUP spdIpsecSystemPolicyNameGroup
    DESCRIPTION
        "This group is mandatory for IPsec Policy
        implementations that support a system policy group
        name."

    GROUP spdCompoundFilterGroup
    DESCRIPTION
        "This group is mandatory for IPsec Policy
        implementations that support compound filters."

    GROUP spdIPOffsetFilterGroup
    DESCRIPTION
        "This group is mandatory for IPsec Policy
        implementations that support IP Offset filters.  In
        general, this SHOULD be supported by a compliant IPsec
```

Policy implementation."

GROUP spdTimeFilterGroup

DESCRIPTION

"This group is mandatory for IPsec Policy implementations that support time filters."

GROUP spdIpsoHeaderFilterGroup

DESCRIPTION

"This group is mandatory for IPsec Policy implementations that support IPSO Header filters."

GROUP spdCompoundActionGroup

DESCRIPTION

"This group is mandatory for IPsec Policy implementations that support compound actions."

OBJECT spdEndGroupLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdGroupContComponentType

SYNTAX INTEGER {

rule(2)

}

DESCRIPTION

"Support of the value group(1) is only required for implementations that support Policy Groups within Policy Groups."

OBJECT spdGroupContLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdRuleDefLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdCompFiltLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdSubFiltLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdIpOffFiltLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdTimeFiltLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdIpsoHeadFiltLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdCompActLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT spdSubActLastChanged

MIN-ACCESS not-accessible

DESCRIPTION

"This object not required for compliance."

OBJECT diffServMultiFieldClfrNextFree

MIN-ACCESS not-accessible

DESCRIPTION

"This object is not required for compliance."

::= { spdCompliances 1 }

spdLoggingCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION

"The compliance statement for SNMP entities that support sending notifications when actions are invoked."

MODULE -- This Module

MANDATORY-GROUPS { spdActionLoggingObjectGroup,
 spdActionNotificationGroup }

::= { spdCompliances 2 }

--

```
-- ReadOnly Compliances
--
spdRuleFilterReadOnlyCompliance MODULE-COMPLIANCE
    STATUS          current
    DESCRIPTION
        "The compliance statement for SNMP entities that include
        an IPsec MIB implementation with Endpoint, Rules, and
        filters support.

        If this MIB is implemented without support for read-create
        (i.e., in read-only), it is not in full compliance, but it
        can claim read-only compliance.  Such a device can then be
        monitored, but cannot be configured with this MIB."

MODULE -- This Module
    MANDATORY-GROUPS { spdEndpointGroup,
                        spdGroupContentsGroup,
                        spdRuleDefinitionGroup,
                        spdStaticFilterGroup,
                        spdStaticActionGroup ,
                        diffServMIBMultiFieldClfrGroup }

GROUP spdIpsecSystemPolicyNameGroup
DESCRIPTION
    "This group is mandatory for IPsec Policy
    implementations that support a system policy group
    name."

GROUP spdCompoundFilterGroup
DESCRIPTION
    "This group is mandatory for IPsec Policy
    implementations that support compound filters."

GROUP spdIPOffsetFilterGroup
DESCRIPTION
    "This group is mandatory for IPsec Policy
    implementations that support IP Offset filters.  In
    general, this SHOULD be supported by a compliant IPsec
    Policy implementation."

GROUP spdTimeFilterGroup
DESCRIPTION
    "This group is mandatory for IPsec Policy
    implementations that support time filters."

GROUP spdIpsoHeaderFilterGroup
DESCRIPTION
    "This group is mandatory for IPsec Policy
```

implementations that support IPSO Header filters."

GROUP spdCompoundActionGroup

DESCRIPTION

"This group is mandatory for IPsec Policy implementations that support compound actions."

OBJECT spdCompActExecutionStrategy

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompActLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdCompActRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompActStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompFiltDescription

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompFiltLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdCompFiltLogicType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompFiltRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdCompFiltStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdEgressPolicyGroupName
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdEndGroupLastChanged
DESCRIPTION

"This object is not required for compliance."

OBJECT spdEndGroupName
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdEndGroupRowStatus
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdEndGroupStorageType
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdGroupContComponentName
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdGroupContComponentType
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdGroupContFilter
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdGroupContLastChanged
DESCRIPTION

"This object is not required for compliance."

OBJECT spdGroupContRowStatus
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdGroupContStorageType
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIngressPolicyGroupName
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpOffFiltLastChanged
DESCRIPTION

"This object is not required for compliance."

OBJECT spdIpOffFiltOffset
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpOffFiltRowStatus
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpOffFiltStorageType
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpOffFiltType
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpOffFiltValue
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpsoHeadFiltClassification
MIN-ACCESS read-only
DESCRIPTION

"Write access is not required."

OBJECT spdIpsoHeadFiltLastChanged
DESCRIPTION

"This object is not required for compliance."

OBJECT spdIpsoHeadFiltProtectionAuth
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdIpsoHeadFiltRowStatus
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdIpsoHeadFiltStorageType
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdIpsoHeadFiltType
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefAction
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefAdminStatus
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefDescription
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefFilter
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefFilterNegated
MIN-ACCESS read-only
DESCRIPTION
 "Write access is not required."

OBJECT spdRuleDefLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdRuleDefRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdRuleDefStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubActLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdSubActRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubActStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubActSubActionName

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubFiltLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdSubFiltRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubFiltStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubFiltSubfilter

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdSubFiltSubfilterIsNegated

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltDayOfMonthMask

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltDayOfWeekMask

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltLastChanged

DESCRIPTION

"This object is not required for compliance."

OBJECT spdTimeFiltMonthOfYearMask

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltPeriod

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltRowStatus

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltTimeOfDayMask

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

OBJECT spdTimeFiltStorageType

MIN-ACCESS read-only

DESCRIPTION

"Write access is not required."

::= { spdCompliances 3 }

```
--
--
-- Compliance Groups Definitions
--

--
-- Endpoint, Rule, Filter Compliance Groups
--

spdEndpointGroup OBJECT-GROUP
    OBJECTS {
        spdEndGroupName, spdEndGroupLastChanged,
        spdEndGroupStorageType, spdEndGroupRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy
        Endpoint Table."
    ::= { spdGroups 1 }

spdGroupContentsGroup OBJECT-GROUP
    OBJECTS {
        spdGroupContComponentType, spdGroupContFilter,
        spdGroupContComponentName, spdGroupContLastChanged,
        spdGroupContStorageType, spdGroupContRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy
        Group Contents Table."
    ::= { spdGroups 2 }

spdIpsecSystemPolicyNameGroup OBJECT-GROUP
    OBJECTS {
        spdIngressPolicyGroupName,
        spdEgressPolicyGroupName
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects represent the System
        Policy Group Names."
    ::= { spdGroups 3 }

spdRuleDefinitionGroup OBJECT-GROUP
    OBJECTS {
        spdRuleDefDescription, spdRuleDefFilter,
        spdRuleDefFilterNegated, spdRuleDefAction,
        spdRuleDefAdminStatus, spdRuleDefLastChanged,
```

```
        spdRuleDefStorageType, spdRuleDefRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy Rule
        Definition Table."
    ::= { spdGroups 4 }

spdCompoundFilterGroup OBJECT-GROUP
    OBJECTS {
        spdCompFiltDescription, spdCompFiltLogicType,
        spdCompFiltLastChanged, spdCompFiltStorageType,
        spdCompFiltRowStatus, spdSubFiltSubfilter,
        spdSubFiltSubfilterIsNegated, spdSubFiltLastChanged,
        spdSubFiltStorageType, spdSubFiltRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy
        Compound Filter Table and Sub-Filter Table Group."
    ::= { spdGroups 5 }

spdStaticFilterGroup OBJECT-GROUP
    OBJECTS { spdTrueFilter }
    STATUS current
    DESCRIPTION
        "The static filter group.  Currently this is just a true
        filter."
    ::= { spdGroups 6 }

spdIPOffsetFilterGroup OBJECT-GROUP
    OBJECTS {
        spdIpOffFiltOffset, spdIpOffFiltType,
        spdIpOffFiltValue, spdIpOffFiltLastChanged,
        spdIpOffFiltStorageType, spdIpOffFiltRowStatus
    }

    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy IP
        Offset Filter Table."
    ::= { spdGroups 7 }

spdTimeFilterGroup OBJECT-GROUP
    OBJECTS {
        spdTimeFiltPeriod,
        spdTimeFiltMonthOfYearMask, spdTimeFiltDayOfMonthMask,
        spdTimeFiltDayOfWeekMask, spdTimeFiltTimeOfDayMask,
```

```

        spdTimeFiltLastChanged,
        spdTimeFiltStorageType, spdTimeFiltRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy Time
        Filter Table."
    ::= { spdGroups 8 }

spdIpsoHeaderFilterGroup OBJECT-GROUP
    OBJECTS {
        spdIpsoHeadFiltType, spdIpsoHeadFiltClassification,
        spdIpsoHeadFiltProtectionAuth, spdIpsoHeadFiltLastChanged,
        spdIpsoHeadFiltStorageType, spdIpsoHeadFiltRowStatus
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy IPSO
        Header Filter Table."
    ::= { spdGroups 9 }

--
-- action compliance groups
--

spdStaticActionGroup OBJECT-GROUP
    OBJECTS {
        spdDropAction, spdAcceptAction,
        spdDropActionLog, spdAcceptActionLog
    }
    STATUS current
    DESCRIPTION
        "This group is made up of objects from the IPsec Policy
        Static Actions."
    ::= { spdGroups 10 }

spdCompoundActionGroup OBJECT-GROUP
    OBJECTS {
        spdCompActExecutionStrategy, spdCompActLastChanged,
        spdCompActStorageType,

        spdCompActRowStatus, spdSubActSubActionName,
        spdSubActLastChanged, spdSubActStorageType,
        spdSubActRowStatus
    }
    STATUS current
    DESCRIPTION
        "The IPsec Policy Compound Action Table and Actions In

```

```
Compound Action Table Group."  
 ::= { spdGroups 11 }
```

```
spdActionLoggingObjectGroup OBJECT-GROUP
```

```
OBJECTS {  
    spdActionExecuted,  
    spdIPEndpointAddType,    spdIPEndpointAddress,  
    spdIPSourceType,        spdIPSourceAddress,  
    spdIPDestinationType,   spdIPDestinationAddress,  
    spdPacketDirection,     spdPacketPart  
}  
STATUS current  
DESCRIPTION  
    "This group is made up of all the Notification objects for  
    this MIB."  
 ::= { spdGroups 12 }
```

```
spdActionNotificationGroup NOTIFICATION-GROUP
```

```
NOTIFICATIONS {  
    spdActionNotification,  
    spdPacketNotification  
}  
STATUS current  
DESCRIPTION  
    "This group is made up of all the Notifications for this MIB."  
 ::= { spdGroups 13 }
```

```
END
```


7. Security Considerations

7.1. Introduction

This document defines a MIB module used to configure IPsec policy services. Since IPsec provides network security services, all of its configuration data (e.g., this entire MIB) SHOULD be as secure or more secure than any of the security services IPsec provides. There are two main threats you need to protect against when configuring IPsec devices.

1. **Malicious Configuration:** This MIB configures network security services. If an attacker has SET access to any part of this MIB, the network security services configured by this MIB SHOULD be considered broken. The network data sent through the associated gateway should no longer be considered as protected by IPsec (i.e., it is no longer confidential or authenticated). Therefore, only the official administrators SHOULD be allowed to configure a device. In other words, administrators' identities SHOULD be authenticated and their access rights checked before they are allowed to do device configuration. The support for SET operations to the SPD MIB in a non-secure environment, without proper protection, will invalidate the security of the network traffic affected by the SPD MIB.
2. **Disclosure of Configuration:** In general, malicious parties SHOULD NOT be able to read security configuration data while the data is in network transit. An attacker reading the configuration data may be able to find misconfigurations in the MIB that enable attacks to the network or to the configured node. Since this entire MIB is used for security configuration, it is highly RECOMMENDED that only authorized administrators are allowed to view data in this MIB. In particular, malicious users SHOULD be prevented from reading SNMP packets containing this MIB's data. SNMP GET data SHOULD be encrypted when sent across the network. Also, only authorized administrators SHOULD be allowed SNMP GET access to any of the MIB objects.

SNMP versions prior to SNMPv3 do not include adequate security. Even if the network itself is secure (e.g., by using IPsec), earlier versions of SNMP have virtually no control as to who on the secure network is allowed to access (i.e., read/change/create/delete) the objects in this MIB module.

It is RECOMMENDED that implementers use the security features as provided by the SNMPv3 framework (see [RFC3410], section 8), including full support for the SNMPv3 cryptographic mechanisms (for authentication and privacy).

Further, deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED. Instead, it is RECOMMENDED to deploy SNMPv3 and to enable cryptographic security. It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB module is properly configured to give access to the objects only to those principals (users) that have legitimate rights to GET or SET (change/create/delete) them.

Therefore, when configuring data in the IPSEC-SPD-MIB, you SHOULD use SNMP version 3. The rest of this discussion assumes the use of SNMPv3. This is a real strength, because it allows administrators the ability to load new IPsec configuration on a device and keep the conversation private and authenticated under the protection of SNMPv3 before any IPsec protections are available. Once initial establishment of IPsec configuration on a device has been achieved, it would be possible to set up IPsec SAs to then also provide security and integrity services to the configuration conversation. This may seem redundant at first, but will be shown to have a use for added privacy protection below.

7.2. Protecting against Unauthenticated Access

The current SNMPv3 User Security Model provides for key-based user authentication. Typically, keys are derived from passwords (but are not required to be), and the keys are then used in Hashed Message Authentication Code (HMAC) algorithms (currently, MD5 and SHA-1 HMACs are defined) to authenticate all SNMP data. Each SNMP device keeps a (configured) list of users and keys. Under SNMPv3 user keys may be updated as often as an administrator cares to have users enter new passwords. But Perfect Forward Secrecy for user keys in SNMPv3 is not yet provided by standards track documents, although RFC2786 defines an experimental method of doing so.

7.3. Protecting against Involuntary Disclosure

While sending IPsec configuration data to a Policy Enforcement Point (PEP), there are a few critical parameters that MUST NOT be observed by third parties. Specifically, except for public keys, keying information MUST NOT be allowed to be observed by third parties. This includes IKE Pre-Shared Keys and possibly the private key of a public/private key pair for use in a PKI. Were either of those parameters to be known to a third party, they could then impersonate the device to other IKE peers. Aside from those critical parameters, policy administrators have an interest in not divulging any of their policy configuration. Any knowledge about a device's configuration could help an unfriendly party compromise that device. SNMPv3 offers privacy security services, but at the time this document was written, the only standardized encryption algorithm supported by SNMPv3 is the

DES encryption algorithm. Support for other (stronger) cryptographic algorithms is in the works and may be completed by the time you read this. As of October 2006, there is a stronger standards track algorithm: AES [RFC3826]. When configuring the IPsec policy using this MIB, policy administrators SHOULD use a privacy security service that is at least as strong as the desired IPsec policy, e.g., If an administrator were to use this MIB to configure an IPsec connection that utilizes a AES algorithms, the SNMP communication configuring the connection SHOULD be protected by an algorithm as strong or stronger than the AES algorithm.

7.4. Bootstrapping Your Configuration

Most vendors will not ship new products with a default SNMPv3 user/password pair, but it is possible. If a device does ship with a default user/password pair, policy administrators SHOULD either change the password or configure a new user, deleting the default user (or, at a minimum, restrict the access of the default user). Most SNMPv3 distributions should, hopefully, require an out-of-band initialization over a trusted medium, such as a local console connection.

8. IANA Considerations

Only two IANA considerations exist for this document. The first is just the node number allocation of the IPSEC-SPD-MIB itself within the MIB-2 tree. This is listed in the MIB definition in Section 6.

The IPSEC-SPD-MIB also allows for extension action MIBs. Although additional actions are not required to use it, the node `spdActions` is allocated as a subtree under which IANA can assign additional actions.

The second IANA consideration is that IANA would be responsible for creating a new subregistry for and assigning nodes under the `spdActions` subtree. This tree should have a prefix of `iso.org.dod.internet.mgmt.mib-2.spdMIB.spdActions` and be listed similar to the following:

Decimal	Name	Description	References
-----	----	-----	-----

A documented specification is required in order to assign a number. The action and it's meaning can be specified in an RFC or in another publicly available reference. The specification should have sufficient detail that interoperability between independent implementations is possible. The product of the IETF or of another standards body is acceptable or an assignment can be accepted under

the advice of a "designated expert". (contact IANA for the current expert)

9. Acknowledgments

Many people contributed thoughts and ideas that influenced this MIB module. Some special thanks are in order to the following people:

Lindy Foster	(Sparta, Inc.)
John Gillis	(ADC)
Roger Hartmuller	(Sparta, Inc.)
Harrie Hazewinkel	
Jamie Jason	(Intel Corporation)
David Partain	(Ericsson)
Lee Rafalow	(IBM)
Jon Saperia	(JDS Consulting)
Eric Vyncke	(Cisco Systems)

10. References

10.1. Normative References

- [RFC1108] Kent, S., "U.S. Department of Defense Security Options for the Internet Protocol", RFC 1108, November 1991.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2401] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

- [RFC3060] Moore, B., Ellessen, E., Strassner, J., and A. Westerinen, "Policy Core Information Model -- Version 1 Specification", RFC 3060, February 2001.
- [RFC3289] Baker, F., Chan, K., and A. Smith, "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC3585] Jason, J., Rafalow, L., and E. Vyncke, "IPsec Configuration Policy Information Model", RFC 3585, August 2003.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.
- [RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.
- [RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

10.2. Informative References

- [IPsec-ACTION] Baer, M., Charlet, R., Hardaker, W., Story, R., and C. Wang, "IPsec Security Policy IPsec Action MIB", Work in Progress, October 2006.
- [IKE-ACTION] Baer, M., Charlet, R., Hardaker, W., Story, R., and C. Wang, "IPsec Security Policy IKE Action MIB", Work in Progress, October 2006.
- [IPPMWP] Lortz, V. and L. Rafalow, "IPsec Policy Model White Paper", November 2000.
- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3826] Blumenthal, U., Maino, F., and K. McCloghrie, "The Advanced Encryption Standard (AES) Cipher Algorithm in the SNMP User-based Security Model", RFC 3826, June 2004.

Authors' Addresses

Michael Baer
Sparta, Inc.
P.O. Box 72682
Davis, CA 95617
US

EMail: baerm@tislabs.com

Ricky Charlet
Self

EMail: rcharlet@alumni.calpoly.edu

Wes Hardaker
Sparta, Inc.
P.O. Box 382
Davis, CA 95617
US

Phone: +1 530 792 1913
EMail: hardaker@tislabs.com

Robert Story
Revelstone Software
PO Box 1812
Tucker, GA 30085
US

EMail: rstory@ipsp.revelstone.com

Cliff Wang
ARO
4300 S. Miami Blvd
Durham, NC 27703
US

EMail: cliffwangmail@yahoo.com

Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

