Network Working Group
Request for Comments: 5190
Category: Standards Track

J. Quittek
M. Stiemerling
NEC
P. Srisuresh
Kazeon Systems

March 2008

Definitions of Managed Objects for Middlebox Communication

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.

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a set of managed objects that allow configuring middleboxes, such as firewalls and network address translators, in order to enable communication across these devices. The definitions of managed objects in this documents follow closely the MIDCOM semantics defined in RFC 5189.

Table of Contents

1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. In particular, it describes a set of managed objects that allow controlling middleboxes.

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].

2. 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].

3. Overview

The managed objects defined in this document serve for controlling firewalls and Network Address Translators (NATs). As defined in [RFC3234], firewalls and NATs belong to the group of middleboxes. A middlebox is a device on the datagram path between source and destination, which performs other functions than just IP routing. As outlined in [RFC3303], firewalls and NATs are potential obstacles to packet streams, for example, if dynamically negotiated UDP or TCP port numbers are used, as in many peer-to-peer communication applications.

As one possible solution for this problem, the IETF MIDCOM working group defined a framework [RFC3303], requirements [RFC3304], and protocol semantics [RFC5189] for communication between applications and middleboxes acting as firewalls, NATs, or a combination of both. The MIDCOM architecture and framework define a model in which trusted third parties can be delegated to assist middleboxes in performing their operations, without requiring application intelligence being embedded in the middleboxes. This trusted third party is referred to as the MIDCOM agent. The MIDCOM protocol is defined between a MIDCOM agent and a middlebox.

The managed objects defined in this document can be used for dynamically configuring middleboxes on the datagram path to permit datagrams traversing the middleboxes. This way, applications can, for example, request pinholes at firewalls and address bindings at NATs.

Besides managed objects for controlling the middlebox operation, this document also defines managed objects that provide information on middlebox resource usage (such as firewall pinholes, NAT bindings, NAT sessions, etc.) affected by requests.

Since firewalls and NATs are critical devices concerning network security, security issues of middlebox communication need to be considered very carefully.

3.1. Terminology

The terminology used in this document is fully aligned with the terminology defined in [RFC5189] except for the term 'MIDCOM agent'. For this term, there is a conflict between the MIDCOM terminology and the SNMP terminology. The roles of entities participating in SNMP communication are called 'manager' and 'agent' with the agent acting as server for requests from the manager. This use of the term 'agent' is different from its use in the MIDCOM framework: The SNMP manager corresponds to the MIDCOM agent and the SNMP agent corresponds to the MIDCOM middlebox, also called MIDCOM server. In order to avoid confusion in this document specifying a MIB module, we replace the term 'MIDCOM agent' with 'MIDCOM client'. Whenever the term 'agent' is used in this document, it refers to the SNMP agent. Figure 1 sketches the entities of MIDCOM in relationship to SNMP manager and SNMP agent.

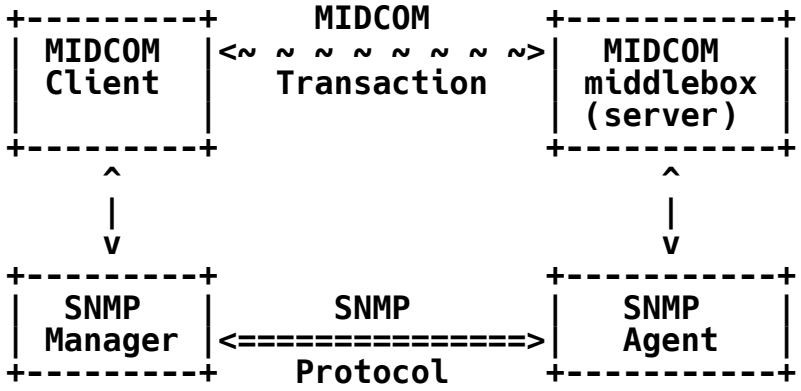


Figure 1: Mapping of MIDCOM to SNMP

4. Realizing the MIDCOM Protocol with SNMP

In order to realize middlebox communication as described in [RFC5189], several aspects and properties of the MIDCOM protocol need to be mapped to SNMP capabilities and expressed in terms of the Structure of Management Information version 2 (SMIv2).

Basic concepts to be mapped are MIDCOM sessions and MIDCOM transactions. For both, access control policies need to be supported.

4.1. MIDCOM Sessions

SNMP has no direct support for sessions. Therefore, they need to be modeled. A MIDCOM session is stateful and has a context that is valid for several transactions. For SNMP, a context is valid for a single transaction only, for example, covering just a single request/reply pair of messages.

Properties of sessions that are utilized by the MIDCOM semantics and not available in SNMP need to be modeled. Particularly, the middlebox needs to be able to authenticate MIDCOM clients, authorize access to policy rules, and send notification messages concerning policy rules to MIDCOM clients participating in a session. In the MIDCOM-MIB module, authentication and access control are performed on a per-message basis using an SNMPv3 security model, such as the User-based Security Model (USM) [RFC3414], for authentication, and the View-based Access Control Model (VACM) [RFC3415] for access control. Sending notifications to MIDCOM clients is controlled by access control models such as VACM and a mostly static configuration of objects in the SNMP-TARGET-MIB [RFC3413] and the SNMP-NOTIFICATION-MIB [RFC3413].

This session model is static except that the MIDCOM client can switch on and off the generation of SNMP notifications that the middlebox sends. Recommended configurations of VACM and the SNMP-TARGET-MIB and the SNMP-NOTIFICATION-MIB that can serve for modeling a session are described in detail in section 6.

4.1.1. Authentication and Authorization

MIDCOM sessions are required for providing authentication, authorization, and encryption for messages exchanged between a MIDCOM client and a middlebox. SNMPv3 provides these features on a per-message basis instead of a per-session basis applying a security model and an access control model, such as USM and VACM. Per-message

security mechanisms can be considered as overhead compared to per-session security mechanisms, but it certainly satisfies the security requirements of middlebox communication.

For each authenticated MIDCOM client, access to the MIDCOM-MIB, particularly to policy rules, should be configured as part of the VACM configuration of the SNMP agent.

4.2. MIDCOM Transactions

[RFC5189] defines the MIDCOM protocol semantics in terms of transactions and transaction parameters. Transactions are grouped into request-reply transactions and asynchronous transactions.

SNMP offers simple transactions that in general cannot be mapped one-to-one to MIDCOM transactions. This section describes how the MIDCOM-MIB module implements MIDCOM transactions using SNMP transactions. The concerned MIDCOM transactions are asynchronous transactions and request-reply transactions. Within the set of request-reply transactions, we distinguish configuration transactions and monitoring transactions, because they are implemented in slightly different ways by using SNMP transactions.

The SNMP terminology as defined in [RFC3411] does not use the concept of transactions, but of SNMP operations. For the considerations in this section, we use the terms SNMP GET transaction and SNMP SET transaction. An SNMP GET transaction consists of an SNMP Read Class operation and an SNMP Response Class operation. An SNMP SET transaction consists of an SNMP Write Class operation and an SNMP Response Class operation.

4.2.1. Asynchronous Transactions

Asynchronous transactions can easily be modeled by SNMP Notification Class operations. An asynchronous transaction contains a notification message with one to three parameters. The message can be realized as an SNMP Notification Class operation with the parameters implemented as managed objects contained in the notification.

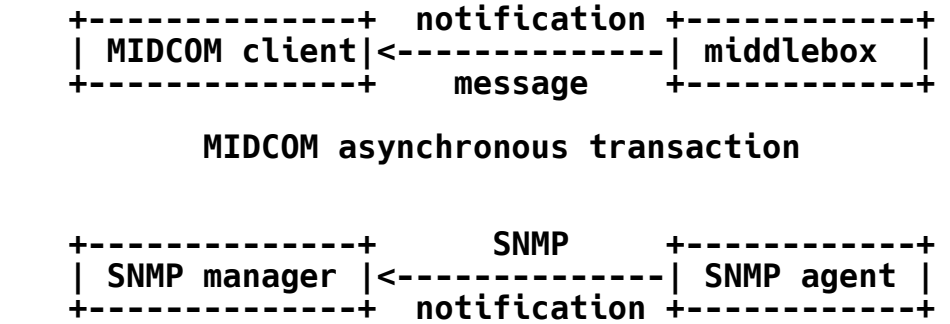


Figure 2: MIDCOM asynchronous transaction mapped to SNMP Notification Class operation

One of the parameters is the transaction identifier that should be unique per middlebox. It does not have to be unique for all notifications sent by the particular SNMP agent, but for all sent notifications that are defined by the MIDCOM-MIB module.

Note that SNMP notifications are usually sent as unreliable UDP packets and may be dropped before they reach their destination. If a MIDCOM client is expecting an asynchronous notification on a specific transaction, it would be the job of the MIDCOM client to poll the middlebox periodically and monitor the transaction in case notifications are lost along the way.

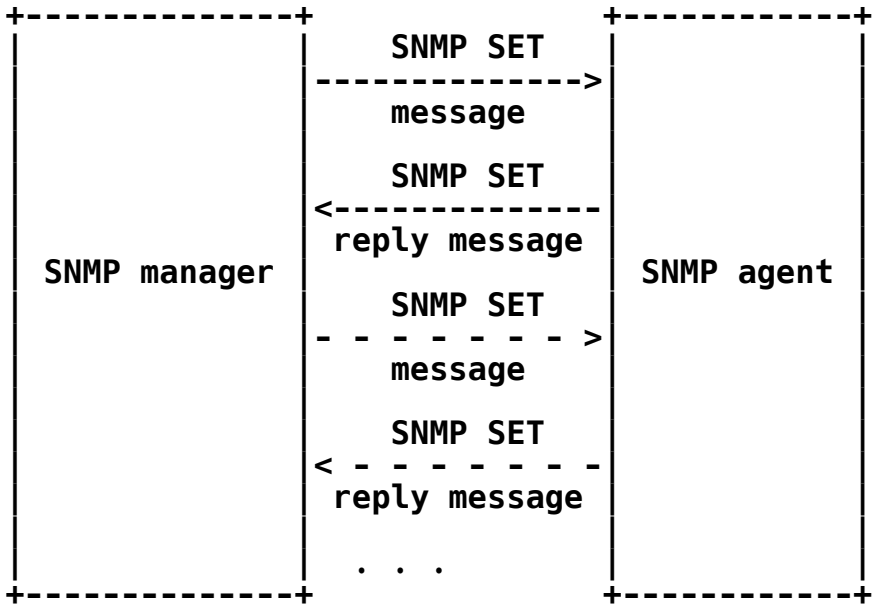
4.2.2. Configuration Transactions

All request-reply transactions contain a request message, a reply message, and potentially also a set of notifications. In general, they cannot be modeled by just having a single SNMP message per MIDCOM message, because some of the MIDCOM messages carry a large set of parameters that do not necessarily fit into an SNMP message consisting of a single UDP packet only.

For configuration transactions, the MIDCOM request message can be modeled by one or more SNMP SET transactions. The action of sending the MIDCOM request to the middlebox is realized by writing the parameters contained in the message to managed objects at the SNMP agent. If necessary, the SNMP SET transaction includes creating these managed objects. If not all parameters of the MIDCOM request message can be set by a single SNMP SET transaction, then more than one SET transaction is used; see Figure 3. Completion of the last of the SNMP transactions indicates that all required parameters are set and that processing of the MIDCOM request message can start at the middlebox.

Please note that a single SNMP SET transaction consists of an SNMP SET request message and an SNMP SET reply message. Both are sent as unreliable UDP packets and may be dropped before they reach their destination. If the SNMP SET request message or the SNMP reply message is lost, then the SNMP manager (the MIDCOM client) needs to take action, for example, by just repeating the SET transaction or by first checking the success of the initial write transaction with an SNMP GET transaction and then only repeating the SNMP SET transaction if necessary.

MIDCOM request message

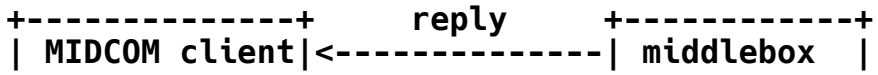


Implementation of MIDCOM request message by one or more SNMP SET transactions

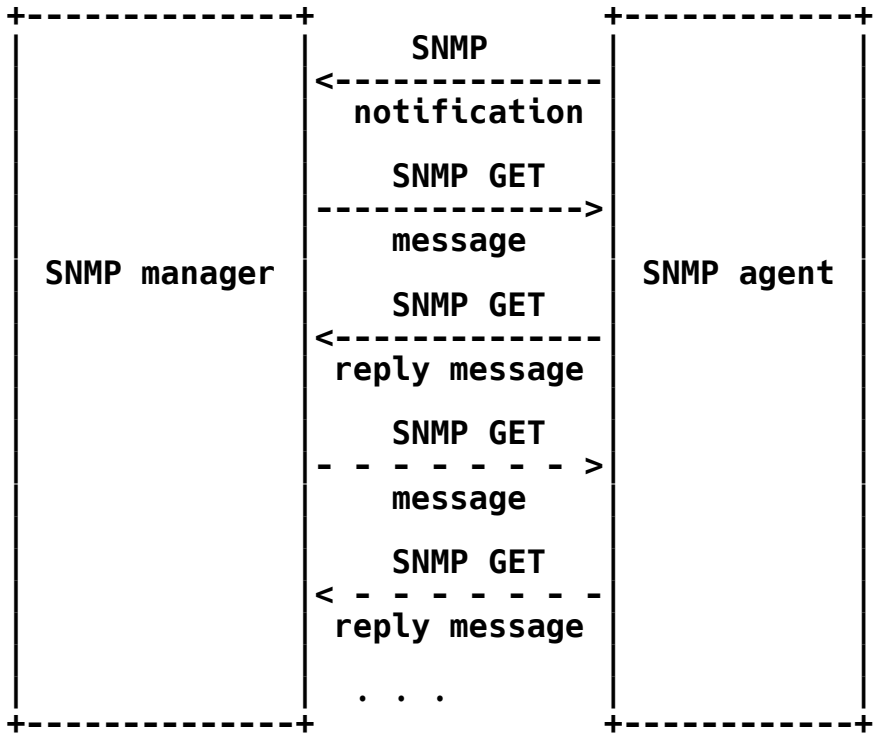
Figure 3: MIDCOM request message mapped to SNMP SET transactions

The MIDCOM reply message can be modeled in two ways. The first way is an SNMP Notification Class operation optionally followed by one or more SNMP GET transactions as shown in Figure 4. The MIDCOM server informs the MIDCOM client about the end of processing the request by sending an SNMP notification. If possible, the SNMP notification

carries all reply parameters. If this is not possible, then the SNMP manager has to perform additional SNMP GET transactions as long as necessary to receive all of the reply parameters.



MIDCOM reply message



Implementation of MIDCOM reply message
by an SNMP notification
and one or more SNMP GET transactions

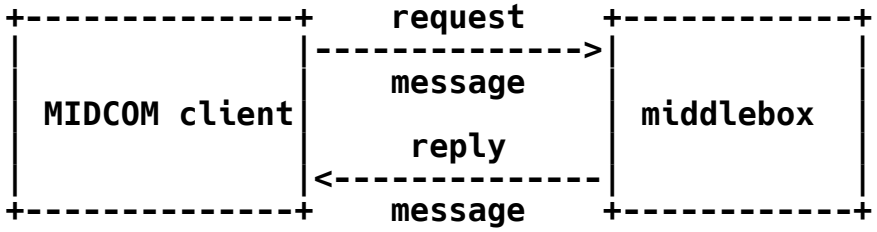
Figure 4: MIDCOM reply message
mapped to SNMP notification and optional GET transactions

The second way replaces the SNMP Notification Class operation by a polling operation of the SNMP manager. The manager polls status information at the SNMP agent using SNMP GET transactions until it detects the end of the processing of the request. Then it uses one or more SNMP GET transactions to receive all of the reply parameters. Note that this second way requires more SNMP operations, but is more

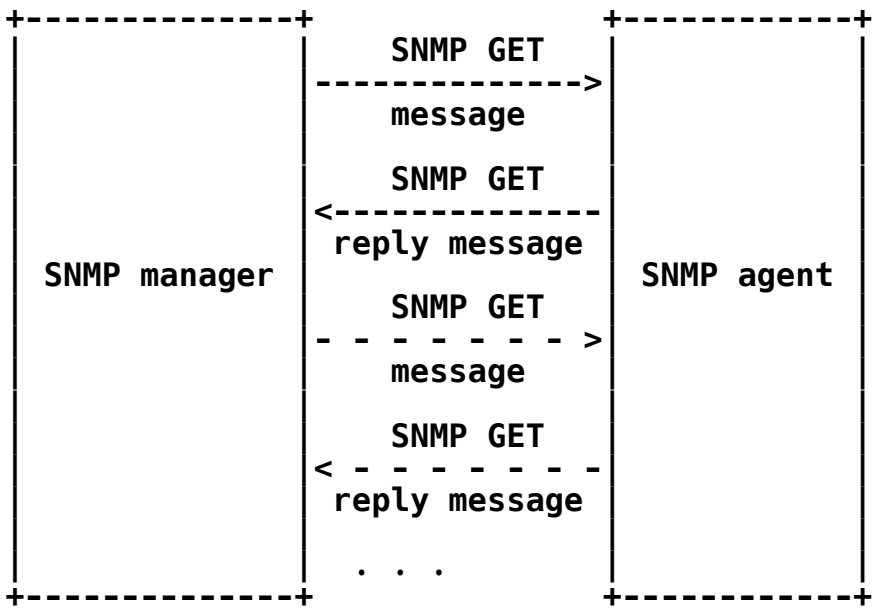
reliable than the first way using an SNMP Notification Class operation.

4.2.3. Monitoring Transactions

The realization of MIDCOM monitoring transactions in terms of SNMP transactions is simpler. The request message is very short and just specifies a piece of information that the MIDCOM client wants to retrieve.



MIDCOM monitoring transaction



Implementation of MIDCOM monitoring transaction by one or more SNMP GET messages

Figure 5: MIDCOM monitoring transaction mapped to SNMP GET transactions

Since monitoring is a strength of SNMP, there are sufficient means to realize MIDCOM monitoring transactions simpler than MIDCOM configuration transactions.

All MIDCOM monitoring transactions can be realized as a sequence of SNMP GET transactions. The number of SNMP GET transactions required depends on the amount of information to be retrieved.

4.2.4. Atomicity of MIDCOM Transactions

Given the realizations of MIDCOM transactions by means of SNMP transactions, atomicity of the MIDCOM transactions is not fully guaranteed anymore. However, this section shows that atomicity provided by the MIB module specified in section 9 is still sufficient for meeting the MIDCOM requirements specified in [RFC3304].

4.2.4.1. Asynchronous MIDCOM Transactions

There are two asynchronous MIDCOM transactions: Asynchronous Session Termination (AST) and Asynchronous Policy Rule Event (ARE). The very static realization of MIDCOM sessions in the MIDCOM-MIB, as described by section 4.1, does not anymore support the asynchronous termination of a session. Therefore, the AST transaction is not modeled. For the ARE, atomicity is maintained, because it is modeled by a single atomic SNMP notification transaction.

In addition, the MIDCOM-MIB supports an Asynchronous Group Event transaction, which is an aggregation of a set of ARE transactions. Also, this MIDCOM transaction is implemented by a single SNMP transaction.

4.2.4.2. Session Establishment and Termination Transactions

The MIDCOM-MIB models MIDCOM sessions in a very static way. The only dynamic actions within these transactions are enabling and disabling the generation of SNMP notifications at the SNMP agent.

For the Session Establishment (SE) transaction, the MIDCOM client first reads the middlebox capabilities. It is not relevant whether or not this action is atomic because a dynamic change of the middlebox capabilities is not to be expected. Therefore, also non-atomic implementations of this action are acceptable.

Then, the MIDCOM agent needs to enable the generation of SNMP notifications at the middlebox. This can be realized by writing to a single managed object in the SNMP-NOTIFICATION-MIB [RFC3413]. But even other implementations are acceptable, because atomicity is not required for this step.

For the Session Termination (ST) transaction, the only required action is disabling the generation of SNMP notifications at the middlebox. As for the SE transaction, this action can be realized atomically by using the SNMP-NOTIFICATION-MIB, but also other implementations are acceptable because atomicity is not required for this action.

4.2.4.3. Monitoring Transactions

Potentially, the monitoring transactions Policy Rule List (PRL), Policy Rule Status (PRS), Group List (GL), and Group Status (GS) are not atomic, because these transactions may be implemented by more than one SNMP GET operation.

The problem that might occur is that while the monitoring transaction is performed, the monitored items may change. For example, while reading a long list of policies, new policies may be added and already read policies may be deleted. This is not in line with the protocol semantics. However, it is not in direct conflict with the MIDCOM requirement requesting the middlebox state to be stable and known by the MIDCOM client, because the middlebox notifies the MIDCOM client on all changes to its state that are performed during the monitoring transaction by sending notifications.

If the MIDCOM client receives such a notification while performing a monitoring transaction (or shortly after completing it), the MIDCOM client can then either repeat the monitoring transaction or integrate the result of the monitoring transaction with the information received via notifications during the transaction. In both cases, the MIDCOM client will know the state of the middlebox.

4.2.4.4. Lifetime Change Transactions

For the policy Rule Lifetime Change (RLC) transaction and the Group Lifetime Change (GLC) transaction, atomicity is maintained. They both have very few parameters for the request message and the reply message. The request parameters can be transmitted by a single SNMP SET request message, and the reply parameters can be transmitted by a single SNMP notification message. In order to prevent idempotency problems by retransmitting an SNMP request after a lost SNMP reply, it is RECOMMENDED that either snmpSetSerialNo (see [RFC3418]) is included in the corresponding SNMP SET request or the value of the SNMP retransmission timer be lower than the smallest requested lifetime value. The same recommendation applies to the smallest requested value for the midcomRuleStorageTime. MIDCOM client implementations MAY completely avoid this problem by configuring their SNMP stack such that no retransmissions are sent.

4.2.4.5. Transactions Establishing New Policy Rules

Analogous to the monitoring transactions, the atomicity may not be given for Policy Reserve Rule (PRR) and Policy Enable Rule (PER) transactions. Both transactions are potentially implemented using more than one SNMP SET operation and GET operation for obtaining transaction reply parameters. The solution for this loss of atomicity is the same as for the monitoring transactions.

There is an additional atomicity problem for PRR and PER. If transferring request parameters requires more than a single SET operation, then there is the potential problem that multiple MIDCOM clients sharing the same permissions are able to access the same policy rule. In this case, a client could alter request parameters already set by another client before the first client could complete the request. However, this is acceptable since usually only one agent is creating a policy rule and filling it subsequently. It can also be assumed that in most cases where clients share permissions, they act in a more or less coordinated way avoiding such interferences.

All atomicity problems caused by using multiple SNMP SET transactions for implementing the MIDCOM request message can be avoided by transferring all request parameters with a single SNMP SET transaction.

4.2.5. Access Control

Since SNMP does not offer per-session authentication and authorization, authentication and authorization are performed per SNMP message sent from the MIDCOM client to the middlebox.

For each transaction, the MIDCOM client has to authenticate itself as an authenticated principal, such as a USM user. Then, the principal's access rights to all resources affected by the transaction are checked. Access right control is realized by configuring the access control mechanisms, such as VACM, at the SNMP agent.

4.3. Access Control Policies

Potentially, a middlebox has to control access for a large set of MIDCOM clients and to a large set of policy rules configuring firewall pinholes and NAT bindings. Therefore, it can be beneficial to use access control policies for specifying access control rules. Generating, provisioning, and managing these policies are out of scope of this MIB module.

However, if such an access control policy system is used, then the SNMP agent acts as a policy enforcement point. An access control policy system must transform all active policies into configurations of, for example, the SNMP agent's View-based Access Control Model (VACM).

The mechanisms of access control models, such as VACM, allow an access control policy system to enforce MIDCOM client authentication rules and general access control of MIDCOM clients to middlebox control.

The mechanisms of VACM can be used to enforce access control of authenticated clients to MIDCOM-MIB policy rules based on the concept of ownership. For example, an access control policy can specify that MIDCOM-MIB policy rules owned by user A cannot be accessed at all by user B, can be read by user C, and can be read and modified by user D.

Further access control policies can control access to concrete middlebox resources. These are enforced, when a MIDCOM request is processed. For example, an authenticated MIDCOM client may be authorized to request new MIDCOM policies to be established, but only for certain IP address ranges. The enforcement of this kind of policies may not be realizable using available SNMP mechanisms, but needs to be performed by the individual MIB module implementation.

5. Structure of the MIB Module

The MIB module defined in section 9 contains three kinds of managed objects:

- Transaction objects
  Transaction objects are required for implementing the MIDCOM protocol requirements defined in [RFC3304] and the MIDCOM protocol semantics defined in [RFC5189].

- Configuration objects
  Configuration objects can be used for retrieving middlebox capability information (mandatory) and for setting parameters of the implementation of transaction objects (optional).

- Monitoring objects
  The optional monitoring objects provide information about used resources and about MIDCOM transaction statistics.

The transaction objects are organized in two tables: the midcomRuleTable and the midcomGroupTable. Entity relationships of

entries of these tables and the midcomResourceTable from the monitoring objects are illustrated by Figure 6.

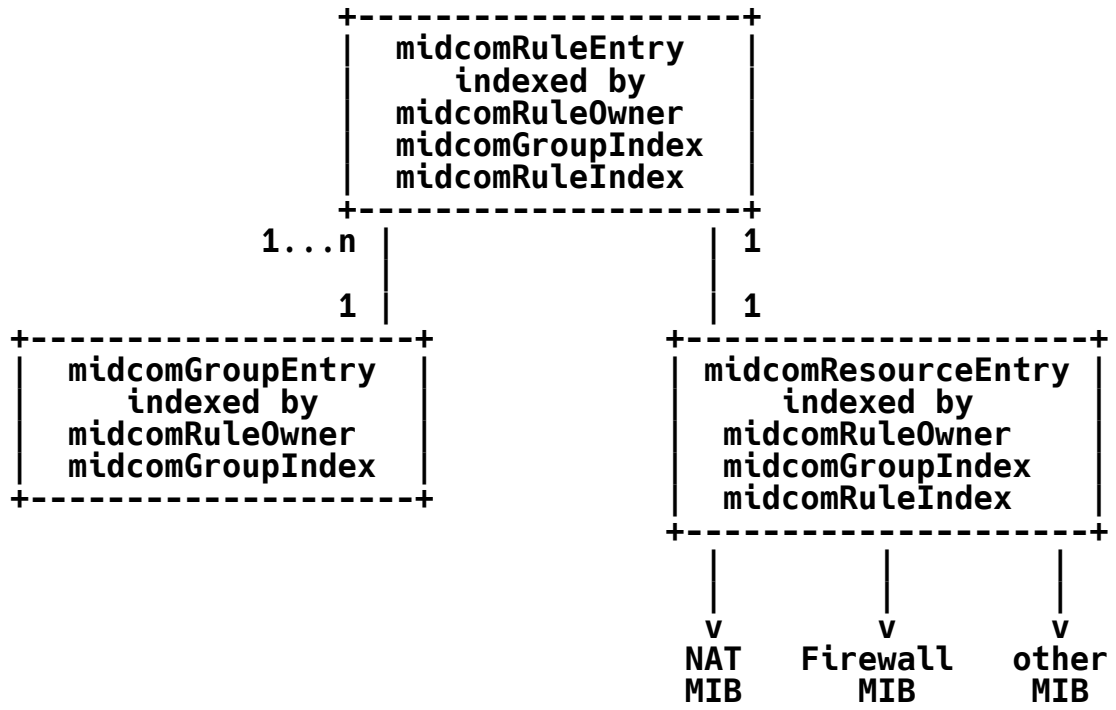


Figure 6: Entity relationships of table entries

A MIDCOM client can create and delete entries in the midcomRuleTable. Entries in the midcomGroupTable are generated automatically as soon as there is an entry in the midcomRuleTable using the midcomGroupIndex. The midcomGroupTable can be used as shortcut for accessing all member rules with a single transaction. MIDCOM clients can group policy rules for various purposes. For example, they can assign a unique value for the midcomGroupIndex to all rules belonging to a single application or an application session served by the MIDCOM agent.

The midcomResourceTable augments the midcomRuleTable by information on the relationship of entries of the midcomRuleTable to resources listed in other MIB modules, such as the NAT-MIB [RFC4008].

5.1. Transaction Objects

The transaction objects are structured according to the MIDCOM semantics described in [RFC5189] into two subtrees, one for policy rule control and one for policy rule group control.

5.1.1. midcomRuleTable

The midcomRuleTable contains information about policy rules including policy rules to be established, policy rules for which establishing failed, established policy rules, and terminated policy rules.

Entries in this table are indexed by the combination of midcomRuleOwner, midcomGroupIndex, and midcomRuleIndex. The midcomRuleOwner is the owner of the rule; the midcomGroupIndex is the index of the group of which the policy rule is a member.

midcomRuleOwner is of type SnmpAdminString, a textual convention that allows for use of the SNMPv3 View-based Access Control Model (VACM [RFC3415]) and allows a management application to identify its entries.

Entries in this table are created by writing to midcomRuleRowStatus. Entries are removed when both their midcomRuleLifetime and midcomRuleStorageTime are timed out by counting down to 0. A MIDCOM client can explicitly remove an entry by setting midcomRuleLifetime and midcomRuleStorageTime to 0.

The table contains the following columnar objects:

- midcomRuleIndex
  The index of this entry must be unique in combination with the midcomRuleOwner and the midcomGroupIndex of the entry.

- midcomRuleAdminStatus
  For establishing a new policy rule, a set of objects in this entry needs to be written first. These objects are the request parameters. Then, by writing either reserve(1) or enable(2) to this object, the MIDCOM-MIB implementation is triggered to start processing the parameters and tries to establish the specified policy rule.

- midcomRuleOperStatus
  This read-only object indicates the current status of the entry. The entry may have an initializing state, it may have a transient state while processing requests, it may have an error state after a request was rejected, it may have a state where a policy rule is established, or it may have a terminated state.

- midcomRuleStorageType
  This object indicates whether or not the policy rule is stored as volatile, non-volatile, or permanent. Depending on the MIDCOM-MIB implementation, this object may be writable.

- midcomRuleStorageTime
  This object indicates how long the entry will still exist after entering an error state or a termination state.

- midcomRuleError
  This object is a string indicating the reason for entering an error state.

- midcomRuleInterface
  This object indicates the IP interface for which enforcement of a policy rule is requested or performed, respectively.

- midcomRuleFlowDirection
  This object indicates a flow direction for which a policy enable rule was requested or established, respectively.

- midcomRuleMaxIdleTime
  This object indicates the maximum idle time of the policy rule in seconds. If no packet to which the policy rule applies passes the middlebox for the time specified by midcomRuleMaxIdleTime, then the policy rule enters a termination state.

- midcomRuleTransportProtocol
  This object indicates a transport protocol for which a policy reserve rule or policy enable rule was requested or established, respectively.

- midcomRulePortRange
  This object indicates a port range for which a policy reserve rule or policy enable rule was requested or established, respectively.

- midcomRuleLifetime
  This object indicates the remaining lifetime of an established policy rule. The MIDCOM client can change the remaining lifetime by writing to it.

Beyond the listed objects, the table contains 10 further objects describing address parameters. They include the IP version, IP address, prefix length and port number for the internal address (A0), inside address (A1), outside address (A2), and external address (A3). These objects serve as parameters specifying a request or an established policy, respectively.

A0, A1, A2, and A3 are address tuples defined according to the MIDCOM semantics [RFC5189]. Each of them identifies either a communication endpoint at an internal or external device or an allocated address at the middlebox.

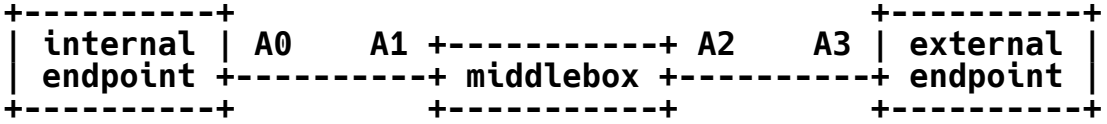


Figure 7: Address tuples A0 - A3

- A0 - internal endpoint: Address tuple A0 specifies a communication endpoint of a device within the internal network, with respect to the middlebox.

- A1 - middlebox inside address: Address tuple A1 specifies a virtual communication endpoint at the middlebox within the internal network. A1 is the destination address for packets passing from the internal endpoint to the middlebox and is the source for packets passing from the middlebox to the internal endpoint.

- A2 - middlebox outside address: Address tuple A2 specifies a virtual communication endpoint at the middlebox within the external network. A2 is the destination address for packets passing from the external endpoint to the middlebox and is the source for packets passing from the middlebox to the external endpoint.

- A3 - external endpoint: Address tuple A3 specifies a communication endpoint of a device within the external network, with respect to the middlebox.

The MIDCOM-MIB requires the MIDCOM client to specify address tuples A0 and A3. This might be a problem for applications that are not designed in a firewall-friendly way. An example is an FTP application that uses the PORT command (instead of the recommended PASV command). The problem only occurs when the middlebox offers twice-NAT functionality, and it can be fixed following recommendations for firewall-friendly communication.

5.1.2. midcomGroupTable

The midcomGroupTable has an entry per existing policy rule group. Entries in this table are created automatically when creating member entries in the midcomRuleTable. Entries are automatically removed from this table when the last member entry is removed from the midcomRuleTable. Entries cannot be created or removed explicitly by the MIDCOM client.

Entries are indexed by the midcomRuleOwner of the rules that belong to the group and by a specific midcomGroupIndex. This allows each midcomRuleOwner to maintain its own independent group namespace.

An entry of the table contains the following objects:

- midcomGroupIndex
  The index of this entry must be unique in combination with the midcomRuleOwner of the entry.

- midcomGroupLifetime
  This object indicates the maximum of the remaining lifetimes of all established policy rules that are members of the group. The MIDCOM client can change the remaining lifetime of all member policies by writing to this object.

5.2. Configuration Objects

The configuration subtree contains middlebox capability and configuration information. Some of the contained objects are (optionally) writable and can also be used for configuring the middlebox service.

The capabilities subtree contains some general capability information and detailed information per supported IP interface. The midcomConfigFirewallTable can be used to configure how the MIDCOM-MIB implementation creates firewall rules in its firewall modules.

Note that typically, configuration objects are not intended to be written by MIDCOM clients. In general, write access to these objects needs to be restricted more strictly than write access to transaction objects.

5.2.1. Capabilities

Information on middlebox capabilities, i.e., capabilities of the MIDCOM-MIB implementation, is provided by the midcomCapabilities subtree of managed objects. The following objects are defined:

- midcomConfigMaxLifetime
  This object indicates the maximum lifetime that this middlebox allows policy rules to have.

- midcomConfigPersistentRules
  This is a boolean object indicating whether or not the middlebox is capable of storing policy rules persistently.

Further capabilities are provided by the midcomConfigIfTable per IP interface. This table contains just two objects. The first one is a BITS object called midcomConfigIfBits containing the following bit values:

- ipv4 and ipv6
  These two bit values provide information on which IP versions are supported by the middlebox at the indexed interface.

- addressWildcards and portWildcards
  These two bit values provide information on wildcarding supported by the middlebox at the indexed interface.

- firewall and nat
  These two bit values provide information on availability of firewall and NAT functionality at the indexed interface.

- portTranslation, protocolTranslation, and twiceNat
  These three bit values provide information on the kind of NAT functionality available at the indexed interface.

- inside
  This bit indicates whether or not the indexed interface is an inside interface with respect to NAT functionality.

The second object, called midcomConfigIfEnabled, indicates whether the middlebox capabilities described by midcomConfigIfBits are available or not available at the indexed IP interface.

The midcomConfigIfTable uses index 0 for indicating capabilities that are available for all interfaces.

5.2.2. midcomConfigFirewallTable

The midcomConfigFirewallTable serves for configuring how policy rules created by MIDCOM clients are realized as firewall rules of a firewall implementation. Particularly, the priority used for MIDCOM-MIB policy rules can be configured. For a single firewall implementation at a particular IP interface, all MIDCOM-MIB policy rules are realized as firewall rules with the same priority. Also, a firewall rule group name can be configured. The table is indexed by the IP interface index.

An entry of the table contains the following objects:

- midcomConfigFirewallGroupId
  This object indicates the firewall rule group to which all firewall rules of the MIDCOM server are assigned.

- midcomConfigFirewallPriority
  This object indicates the priority assigned to all firewall rules of the MIDCOM server.

5.3. Monitoring Objects

The monitoring objects are structured into two subtrees: the resource subtree and the statistics subtree. The resource subtree provides information about which resources are used by which policy rule. The statistics subtree provides statistics about the usage of transaction objects.

5.3.1. midcomResourceTable

Information about resource usage per policy rule is provided by the midcomResourceTable. Each entry in the midcomResourceTable describes resource usage of exactly one policy rule.

Resources are NAT resources and firewall resources, depending on the type of middlebox. Used NAT resources include NAT bindings and NAT sessions. NAT address mappings are not covered. For firewalls, firewall filter rules are considered as resources.

The values provided by the following objects on NAT binds and NAT sessions may refer to the detailed resource usage description in the NAT-MIB module [RFC4008].

The values provided by the following objects on firewall rules may refer to more detailed firewall resource usage descriptions in other MIB modules.

Entries in the midcomResourceTable are only valid if the midcomRuleOperStatus object of the corresponding entry in the midcomRuleTable has a value of either reserved(7) or enabled(8).

An entry of the table contains the following objects:

- midcomRscNatInternalAddrBindMode
  This object indicates whether the binding of the internal address is an address NAT binding or an address-port NAT binding.

- midcomRscNatInternalAddrBindId
  This object identifies the NAT binding for the internal address in the NAT engine.

- midcomRscNatExternalAddrBindMode
  This object indicates whether the binding of the external address is an address NAT binding or an address-port NAT binding.

- midcomRscNatExternalAddrBindId
  This object identifies the NAT binding for the external address in the NAT engine.

- midcomRscNatSessionId1
  This object links to the first NAT session associated with one of the above NAT bindings.

- midcomRscNatSessionId2
  This object links to the optional second NAT session associated with one of the above NAT bindings.

- midcomRscFirewallRuleId
  This object indicates the firewall rule for this policy rule.

The MIDCOM-MIB module does not require a middlebox to implement further specific middlebox (NAT, firewall, etc.) MIB modules as, for example, the NAT-MIB module [RFC4008].

The resource identifiers in the midcomResourceTable may be vendor proprietary in the cases where the middlebox does not implement the NAT-MIB [RFC4008] or a firewall MIB. The MIDCOM-MIB module affects NAT binding and sessions, as well as firewall pinholes. It is intentionally not specified in the MIDCOM-MIB module how these NAT and firewall resources are allocated and managed, since this depends on the MIDCOM-MIB implementation and middlebox's capabilities. However, the midcomResourceTable is useful for understanding which resources are affected by which MIDCOM-MIB transaction.

The midcomResourceTable is beneficial to the middlebox administrator in that the table lists all MIDCOM transactions and the middlebox specific resources to which these transactions refer. For instance, multiple MIDCOM clients might end up using the same NAT bind, yet each MIDCOM client might define a Lifetime parameter and directionality for the bind that is specific to the transaction. MIDCOM-MIB implementations are responsible for impacting underlying middlebox resources so as to satisfy the sometimes overlapping requirements on the same resource from multiple MIDCOM clients.

Managing these resources is not a trivial task for MIDCOM-MIB implementers. It is possible that different MIDCOM-MIB policy rules owned by different MIDCOM clients share a NAT binding or a firewall rule. Then common properties, for example, the lifetime of the resource, need to be managed such that all clients are served well and changes to these resources need to be communicated to all affected clients. Also, dependencies between resources, for example, the precedence order of firewall rules, need to be considered

carefully in order to avoid that different policy rules -- potentially owned by different clients -- influence each other.

MIDCOM clients may use the midcomResourceTable of the MIDCOM-MIB module in conjunction with the NAT-MIB module [RFC4008] to determine which resources of the NAT are used for MIDCOM. The NAT-MIB module stores the configured NAT bindings and sessions, and MIDCOM clients can use the information of the midcomResourceTable to sort out those NAT resources that are used by the MIDCOM-MIB module.

5.3.2. midcomStatistics

The statistics subtree contains a set of non-columnar objects that provide 'MIDCOM protocol statistics', i.e., statistics about the usage of transaction objects.

- midcomCurrentOwners
  This object indicates the number of different values for midcomRuleOwner for all current entries in the midcomRuleTable.

- midcomOwnersTotal
  This object indicates the summarized number of all different values that occurred for midcomRuleOwner in the midcomRuleTable current and in the past.

- midcomTotalRejectedRuleEntries
  This object indicates the total number of failed attempts to create an entry in the midcomRuleTable.

- midcomCurrentRulesIncomplete
  This object indicates the total number of policy rules that have not been fully loaded into a table row of the midcomRuleTable.

- midcomTotalIncorrectReserveRules
  This object indicates the total number of policy reserve rules that were rejected because the request was incorrect.

- midcomTotalRejectedReserveRules
  This object indicates the total number of policy reserve rules that were failed while being processed.

- midcomCurrentActiveReserveRules
  This object indicates the number of currently active policy reserve rules in the midcomRuleTable.

- midcomTotalExpiredReserveRules
  This object indicates the total number of expired policy reserve rules.

- midcomTotalTerminatedOnRqReserveRules
  This object indicates the total number of policy reserve rules that were terminated on request.

- midcomTotalTerminatedReserveRules
  This object indicates the total number of policy reserve rules that were terminated, but not on request.

- midcomTotalIncorrectEnableRules
  This object indicates the total number of policy enable rules that were rejected because the request was incorrect.

- midcomTotalRejectedEnableRules
  This object indicates the total number of policy enable rules that were failed while being processed.

- midcomCurrentActiveEnableRules
  This object indicates the number of currently active policy enable rules in the midcomRuleTable.

- midcomTotalExpiredEnableRules
  This object indicates the total number of expired policy enable rules.

- midcomTotalTerminatedOnRqEnableRules
  This object indicates the total number of policy enable rules that were terminated on request.

- midcomTotalTerminatedEnableRules
  This object indicates the total number of policy enable rules that were terminated, but not on request.

5.4. Notifications

For informing MIDCOM clients about state changes of MIDCOM-MIB implementations, three notifications can be used. They notify the MIDCOM client about state changes of individual policy rules or of groups of policy rules. Different notifications are used for different kinds of transactions.

For asynchronous transactions, unsolicited notifications are used. The only asynchronous transaction that needs to be modeled by the MIDCOM-MIB is the Asynchronous Policy Rule Event (ARE). The ARE may be caused by the expiration of a policy rule lifetime, the expiration of the idle time, or an internal change in policy rule lifetime by the MIDCOM-MIB implementation for whatever reason.

For configuration transactions, solicited notifications are used. This concerns the Policy Reserve Rule (PRR) transaction, the Policy Enable Rule (PER) transaction, the Policy Rule Lifetime Change (RLC) transaction, and the Group Lifetime Change (GLC) transaction.

The separation between unsolicited and solicited notifications gives the implementer of a MIDCOM client some freedom to make design decisions on how to model the MIDCOM reply message as described at the end of section 4.2.2. Depending on the choice, processing of solicited notifications may not be required. In such a case, delivery of solicited notification may be disabled, for example, by an appropriate configuration of the snmpNotifyFilterTable such that solicited notifications are filtered differently to unsolicited notifications.

- midcomUnsolicitedRuleEvent
  This notification can be generated for indicating the change of a policy rule's state or lifetime. It is used for performing the ARE transaction.

- midcomSolicitedRuleEvent
  This notification can be generated for indicating the requested change of a policy rule's state or lifetime. It is used for performing PRR, PER, and RLC transactions.

- midcomSolicitedGroupEvent
  This notification can be generated for indicating the requested change of a policy rule group's lifetime. It is used for performing the GLC transaction.

6. Recommendations for Configuration and Operation

Configuring MIDCOM-MIB security is highly sensitive for obvious reasons. This section gives recommendations for securely configuring the SNMP agent acting as MIDCOM server. In addition, recommendations for avoiding idempotency problems are given and restrictions of MIDCOM-MIB applicability to a special set of applications are discussed.

6.1. Security Model Configuration

Since controlling firewalls and NATs is highly sensitive, it is RECOMMENDED that SNMP Command Responders implementing the MIDCOM-MIB module use the authPriv security level for all users that may access managed objects of the MIDCOM-MIB module.

6.2. VACM Configuration

Entries in the midcomRuleTable and the midcomGroupTable provide information about existing firewall pinholes and/or NAT sessions. They also could be used for manipulating firewall pinholes and/or NAT sessions. Therefore, access control to these objects is essential and should be restrictive.

It is RECOMMENDED that SNMP Command Responders instantiating an implementation of the MIDCOM-MIB module use VACM for controlling access to managed objects in the midcomRuleTable and the midcomGroupTable.

It is further RECOMMENDED that individual MIDCOM clients, acting as SNMP Command Generators, only have access to an entry in the midcomRuleTable, the midcomResourceTable, or the midcomGroupTable, if they created the entry directly in the midcomRuleTable or indirectly in the midcomGroupTable and midcomResourceTable. Exceptions to this recommendation are situations where access by multiple MIDCOM clients to managed objects is explicitly required. One example is fail-over for MIDCOM agents where the stand-by MIDCOM agent needs the same access rights to managed objects as the currently active MIDCOM agent. Another example is a supervisor MIDCOM agent that monitors activities of other MIDCOM agents and/or may be used by network management systems to modify entries in tables of the MIDCOM-MIB.

For this reason, all three tables listed above have the midcomRuleOwner as initial index. It is RECOMMENDED that MIDCOM clients acting as SNMP Command Generator have access to the midcomRuleTable and the midcomGroupTable restricted to entries with the initial index matching either their SNMP securityName or their VACM groupName. It is RECOMMENDED that they do not have access to entries in these tables with initial indices other than their SNMP securityName or their VACM groupName. It is RECOMMENDED that this VACM configuration is applied to read access, write access, and notify access for all objects in the midcomRuleTable and the midcomGroupTable.

Note that less restrictive access rights MAY be granted to other users, for example, to a network management application, that monitors MIDCOM policy rules.

6.3. Notification Configuration

For each MIDCOM client that has access to the midcomRuleTable, a notification target SHOULD be configured at a Command Responder instantiating an implementation of the MIDCOM-MIB. It is RECOMMENDED that such a configuration be retrievable from the Command Responder via the SNMP-TARGET-MIB [RFC3413].

For each entry of the snmpTargetAddrTable that is related to a MIDCOM client, there SHOULD be an individual corresponding entry in the snmpTargetParamsTable.

An implementation of the MIDCOM-MIB SHOULD also implement the SNMP-NOTIFICATION-MIB [RFC3413]. An instance of an implementation of the MIDCOM-MIB SHOULD have an individual entry in the snmpNotifyFilterProfileTable for each MIDCOM client that has access to the midcomRuleTable.

An instance of an implementation of the MIDCOM-MIB SHOULD allow MIDCOM clients to start and stop the generation of notifications targeted at themselves. This SHOULD be realized by giving the MIDCOM clients write access to the snmpNotifyFilterTable. If appropriate entries of the snmpNotifyFilterTable are established in advance, then this can be achieved by granting MIDCOM clients write access only to the columnar object snmpNotifyFilterType.

It is RECOMMENDED that VACM be configured such that each MIDCOM agent can only access entries in the snmpTargetAddrTable, the snmpTargetParamsTable, the snmpNotifyFilterProfileTable, and the snmpFilterTable that concern that particular MIDCOM agent. Typically, read access to the snmpTargetAddrTable, the snmpTargetParamsTable, and the snmpNotifyFilterProfileTable is sufficient. Write access may be required for objects of the snmpFilterTable.

6.4. Simultaneous Access

Situations with two MIDCOM clients simultaneously modifying the same policy rule should be avoided. For each entry in the midcomRuleTable, there should be only one client at a time that modifies it. If two MIDCOM clients share the same midcomRuleOwner index of the midcomRuleTable, then conflicts can be avoided, for example, by

- scheduling access times, as, for example, in the fail-over case;
- using different midcomGroupIndex values per client; or
- using non-overlapping intervals for values of the midcomRuleIndex per client.

6.5. Avoiding Idempotency Problems

As already discussed in section 4.2.4.4, the following recommendation is given for avoiding idempotency problems.

In general, idempotency problems can be solved by including snmpSetSerialNo (see [RFC3418]) in SNMP SET requests.

In case this feature is not used, it is RECOMMENDED that the value of the SNMP retransmission timer of a MIDCOM client (acting as SNMP Command Generator) is lower than the smallest requested value for any rule lifetime or rule idle time in order to prevent idempotency problems with setting midcomRuleLifetime and midcomRuleMaxIdleTime when retransmitting an SNMP SET request after a lost SNMP reply.

MIDCOM client implementations MAY completely avoid this problem by configuring their SNMP stack such that no retransmissions are sent.

Similar considerations apply to MIDCOM-MIB implementations acting as Notification Originator when sending a notification (midcomUnsolicitedRuleEvent, midcomSolicitedRuleEvent or midcomSolicitedGroupEvent) containing the remaining lifetime of a policy rule or a policy rule group, respectively.

6.6. Interface Indexing Problems

A well-known problem of MIB modules is indexing IP interfaces after a re-initialization of the managed device. The index for interfaces provided by the ifTable (see IF-MIB in [RFC2863]) may change during re-initialization, for example, when physical interfaces are added or removed.

The MIDCOM-MIB module uses the interface index for indicating at which interface which policy rule is (or is to be) applied. Also, this index is used for indicating how policy rules are prioritized at certain interfaces. The MIDCOM-MIB module specification requires that information provided is always correct. This implies that after re-initialization, interface index values of policy rules or firewall configurations may have changed even though they still refer to the same interface as before the re-initialization.

MIDCOM client implementations need to be aware of this potential behavior. It is RECOMMENDED that before writing the value or using the value of indices that depend on the ifTable the MIDCOM client checks if the middlebox has been re-initialized recently.

MIDCOM-MIB module implementations MUST track interface changes of IP interface indices in the ifTable. This implies that after a re-initialization of a middlebox, a MIDCOM-MIB implementation MUST make sure that each instance of an interface index in the MIDCOM-MIB tables still points to the same interface as before the re-initialization. For any instance for which this is not possible, all affected entries in tables of the MIDCOM-MIB module MUST be either terminated, disabled, or deleted, as specified in the DESCRIPTION clause of the respective object. This concerns all objects in the MIDCOM-MIB module that are of type InterfaceIndexOrZero.

6.7. Applicability Restrictions

As already discussed in section 5.1.1, the MIDCOM-MIB requires the MIDCOM client to specify address tuples A0 and A3. This can be a problem for applications that do not have this information available when they need to configure the middlebox. For some applications, there are usage scenarios where address information is only available for a single address realm, A0 and A1 in the private realm or A2 and A3 in the public realm. An example is an FTP application using the PORT command (instead of the PASV command). The problem occurs when the middlebox offers twice-NAT functionality.

7. Usage Examples for MIDCOM Transactions

This section presents some examples that explain how a MIDCOM client acting as SNMP manager can use the MIDCOM-MIB module defined in this memo. The purpose of these examples is to explain the steps that are required to perform MIDCOM transactions. For each MIDCOM transaction defined in the MIDCOM semantics [RFC5189], a sequence of SNMP operations that realizes the transaction is described.

The examples described below are recommended procedures for MIDCOM clients. Clients may choose to operate differently.

For example, they may choose not to receive solicited notifications on completion of a transaction, but to poll the MIDCOM-MIB instead until the transaction is completed. This can be achieved by performing step 2 of the SE transaction (see below) differently. The MIDCOM agent then creates an entry in the snmpNotifyFilterTable such that only the midcomUnsolicitedRuleEvent may pass the filter and is sent to the MIDCOM client. In this case, the PER, PRR, and RLC transactions require a polling loop wherever in the example below the MIDCOM client waits for a notification.

7.1. Session Establishment (SE)

The MIDCOM-MIB realizes most properties of MIDCOM sessions in a very static way. Only the generation of notifications targeted at the MIDCOM client is enabled by the client for session establishment.

1. The MIDCOM client checks the middlebox capabilities by reading objects in the midcomCapabilitiesGroup.

2. The MIDCOM client enables generation of notifications on events concerning the policy rules controlled by the client. If the SNMP-NOTIFICATION-MIB is supported as recommended by section 6.3 of this document, then the agent just has to change the value of a object snmpNotifyFilterType in the corresponding entry of the snmpNotifyFilterTable from included(1) to excluded(2).

7.2. Session Termination (ST)

For terminating a session, the MIDCOM client just disables the generation of notifications for this client.

1. The MIDCOM client disables generation of notifications on events concerning the policy rules controlled by the client. If the SNMP-NOTIFICATION-MIB is supported as recommended by section 6.3 of this document, then the agent just has to change the value of a object snmpNotifyFilterType in the corresponding entry of the snmpNotifyFilterTable from included(1) to excluded(2).

7.3. Policy Reserve Rule (PRR)

This example explains steps that may be performed by a MIDCOM client to establish a policy reserve rule.

1. The MIDCOM client creates a new entry in the midcomRuleTable by writing to midcomRuleRowStatus. The chosen value for index object midcomGroupIndex determines the group membership of the created rule. Note that choosing an unused value for midcomGroupIndex creates a new entry in the midcomGroupTable.

2. The MIDCOM client sets the following objects in the new entry of the midcomRuleTable to specify all request parameters of the PRR transaction:

   - midcomRuleMaxIdleTime
   - midcomRuleInterface
   - midcomRuleTransportProtocol
   - midcomRulePortRange
   - midcomRuleInternalIpVersion

- midcomRuleExternalIpVersion
- midcomRuleInternalIpAddr
- midcomRuleInternalIpPrefixLength
- midcomRuleInternalPort
- midcomRuleLifetime

Note that several of these parameters have default values that can be used.

3. The MIDCOM client sets the midcomRuleAdminStatus objects in the new row of the midcomRuleTable to reserve(1).

4. The MIDCOM client awaits a midcomSolicitedRuleEvent notification concerning the new policy rule in the midcomRuleTable. Waiting for the notification is timed out after a pre-selected maximum waiting time. In case of a timeout while waiting for the notification or if the client does not use notifications, the MIDCOM client retrieves the status of the midcomRuleEntry by one or more SNMP GET operations.

5. After receiving the midcomSolicitedRuleEvent notification, the MIDCOM client checks the lifetime value carried by the notification. If it is greater than 0, the MIDCOM client reads all positive reply parameters of the PRR transaction:

   - midcomRuleOutsideIpAddr
   - midcomRuleOutsidePort
   - midcomRuleMaxIdleTime
   - midcomRuleLifetime

   If the lifetime equals 0, then the MIDCOM client reads the midcomRuleOperStatus and the midcomRuleError in order to analyze the failure reason.

6. Optionally, after receiving the midcomSolicitedRuleEvent notification with a lifetime value greater than 0, the MIDCOM client may check the midcomResourceTable for the middlebox resources allocated for this policy reserve rule. Note that PRR does not necessarily allocate any middlebox resource visible in the NAT-MIB module or in a firewall MIB module, since it does a reservation only. If, however, the PRR overlaps with already existing PERs, then the PRR may be related to middlebox resources visible in other MIB modules.

7.4. Policy Enable Rule (PER) after PRR

This example explains steps that may be performed by a MIDCOM client to establish a policy enable rule after a corresponding policy reserve rule was already established.

1. The MIDCOM client sets the following objects in the row of the established PRR in the midcomRuleTable to specify all request parameters of the PER transaction:

   - midcomRuleMaxIdleTime
   - midcomRuleExternalIpAddr
   - midcomRuleExternalIpPrefixLength
   - midcomRuleExternalPort
   - midcomRuleFlowDirection

   Note that several of these parameters have default values that can be used.

2. The MIDCOM client sets the midcomRuleAdminStatus objects in the row of the established PRR in the midcomRuleTable to enable(1).

3. The MIDCOM client awaits a midcomSolicitedRuleEvent notification concerning the new row in the midcomRuleTable. Waiting for the notification is timed out after a pre-selected maximum waiting time. In case of a timeout while waiting for the notification or if the client does not use notifications, the MIDCOM client retrieves the status of the midcomRuleEntry by one or more SNMP GET operations.

4. After receiving the midcomSolicitedRuleEvent notification, the MIDCOM client checks the lifetime value carried by the notification. If it is greater than 0, the MIDCOM client reads all positive reply parameters of the PER transaction:

   - midcomRuleInsideIpAddr
   - midcomRuleInsidePort
   - midcomRuleMaxIdleTime

   If the lifetime equals 0, then the MIDCOM client reads the midcomRuleOperStatus and the midcomRuleError in order to analyze the failure reason.

5. Optionally, after receiving the midcomSolicitedRuleEvent notification with a lifetime value greater than 0, the MIDCOM client may check the midcomResourceTable for the allocated middlebox resources for this policy enable rule.

7.5. Policy Enable Rule (PER) without Previous PRR

This example explains steps that may be performed by a MIDCOM client to establish a policy enable rule for which no PRR transaction has been performed before.

1. Identical to step 1 for PRR (section 7.3).

2. Identical to step 2 for PRR (section 7.3).

3. The MIDCOM client sets the following objects in the new row of the midcomRuleTable to specify all request parameters of the PER transaction:

   - midcomRuleInterface
   - midcomRuleFlowDirection
   - midcomRuleTransportProtocol
   - midcomRulePortRange
   - midcomRuleInternalIpVersion
   - midcomRuleExternalIpVersion
   - midcomRuleInternalIpAddr
   - midcomRuleInternalIpPrefixLength
   - midcomRuleInternalPort
   - midcomRuleExternalIpAddr
   - midcomRuleExternalIpPrefixLength
   - midcomRuleExternalPort
   - midcomRuleLifetime

   Note that several of these parameters have default values that can be used.

4. The MIDCOM client sets the midcomRuleAdminStatus objects in the new row of the midcomRuleTable to enable(1).

5. Identical to step 4 for PRR (section 7.3).

6. After receiving the midcomSolicitedRuleEvent notification, the MIDCOM client checks the lifetime value carried by the notification. If it is greater than 0, the MIDCOM client reads all positive reply parameters of the PRR transaction:

   - midcomRuleInsideIpAddr
   - midcomRuleInsidePort
   - midcomRuleOutsideIpAddr
   - midcomRuleOutsidePort
   - midcomRuleMaxIdleTime

If the lifetime equals 0, then the MIDCOM client reads the midcomRuleOperStatus and the midcomRuleError in order to analyze the failure reason.

7. Optionally, after receiving the midcomSolicitedRuleEvent notification with a lifetime value greater than 0, the MIDCOM client may check the midcomResourceTable for the allocated middlebox resources for this policy enable rule.

7.6. Policy Rule Lifetime Change (RLC)

This example explains steps that may be performed by a MIDCOM client to change the lifetime of a policy rule. Changing the lifetime to 0 implies terminating the policy rule.

1. The MIDCOM client issues a SET request for writing the desired lifetime to the midcomRuleLifetime object in the corresponding row of the midcomRuleTable. This does not have any effect if the lifetime is already expired.

2. The MIDCOM client awaits a midcomSolicitedRuleEvent notification concerning the corresponding row in the midcomRuleTable. Waiting for the notification is timed out after a pre-selected maximum waiting time. In case of a timeout while waiting for the notification or if the client does not use notifications, the MIDCOM client retrieves the status of the midcomRuleEntry by one or more SNMP GET operations.

3. After receiving the midcomSolicitedRuleEvent notification MIDCOM client checks the lifetime value carried by the notification.

7.7. Policy Rule List (PRL)

The SNMP agent can browse the list of policy rules by browsing the midcomRuleTable. For each observed row in this table, the SNMP agent should check the midcomRuleOperStatus in order to find out if the row contains information about an established policy rule or of a rule that is under construction or already terminated.

7.8. Policy Rule Status (PRS)

The SNMP agent can retrieve all status information and properties of a policy rule by reading the managed objects in the corresponding row of the midcomRuleTable.

7.9. Asynchronous Policy Rule Event (ARE)

There are two different triggers for the ARE. It may be triggered by the expiration of a policy rule's lifetime or the expiration of the idle time. But beyond this, the MIDCOM-MIB implementation may terminate a policy rule at any time. In all cases, two steps are required for performing this transaction:

1. The MIDCOM-MIB implementation sends a midcomUnsolicitedRuleEvent notification containing a lifetime value of 0 to the MIDCOM client owning the rule.

2. If the midcomRuleStorageTime object in the corresponding row of the midcomRuleTable has a value of 0, then the MIDCOM-MIB implementation removes the row from the table. Otherwise, it sets in this row the midcomRuleLifetime object to 0 and changes the midcomRuleOperStatus object. If the event was triggered by policy lifetime expiration, then the midcomRuleOperStatus is set to timedOut(9); otherwise, it is set to terminated(11).

7.10. Group Lifetime Change (GLC)

This example explains steps that may be performed by a MIDCOM client to change the lifetime of a policy rule group. Changing the lifetime to 0 implies terminating all member policies of the group.

1. The MIDCOM client issues a SET request for writing the desired lifetime to the midcomGroupLifetime object in the corresponding row of the midcomGroupTable.

2. The MIDCOM client waits for a midcomSolicitedGroupEvent notification concerning the corresponding row in the midcomGroupTable. Waiting for the notification is timed out after a pre-selected maximum waiting time. In case of a timeout while waiting for the notification or if the client does not use notifications, the MIDCOM client retrieves the status of the midcomGroupEntry by one or more SNMP GET operations.

3. After receiving the midcomSolicitedRuleEvent notification, the MIDCOM client checks the lifetime value carried by the notification.

7.11. Group List (GL)

The SNMP agent can browse the list of policy rule groups by browsing the midcomGroupTable. For each observed row in this table, the SNMP agent should check the midcomGroupLifetime in order to find out if the group does contain established policies.

7.12. Group Status (GS)

The SNMP agent can retrieve all member policies of a group by browsing the midcomRuleTable using the midcomGroupIndex of the particular group. For retrieving the remaining lifetime of the group, the SNMP agent reads the midcomGroupLifetime object in the corresponding row of the midcomGroupTable.

8. Usage Examples for Monitoring Objects

This section presents some examples that explain how a MIDCOM client can use the midcomResourceTable to correlate policy rules with the used middlebox resources. One example is given for middleboxes implementing the NAT-MIB and another one is given for firewalls.

8.1. Monitoring NAT Resources

When a rule in the midcomRuleTable is executed, it directly impacts the middlebox resources. The midcomResourceTable provides the information on the relationships between the MIDCOM-MIB policy rules and the middlebox resources used for enforcing these rules.

A MIDCOM-MIB policy rule will cause the creation or modification of up to two NAT bindings and up to two NAT sessions. Two NAT bindings are impacted in the case of a session being subject to twice-NAT. Two NAT bindings may also be impacted when midcomRulePortRange is set to pair(2) in the policy rule. In the majority of cases, where traditional NAT is implemented, only a single NAT binding may be adequate. Note, however, that this BindId is set to 0 if the middlebox is implementing symmetric NAT function. Two NAT sessions are created or modified only when the midcomRulePortRange is set to pair(2) in the policy rule.

When support for the NAT-MIB module is also available at the middlebox, the parameters in the combination of the midcomRuleTable and the midcomResourceTable for a given rule can be used to index the corresponding BIND and NAT session resources effected in the NAT-MIB. These parameters are valuable to monitor the impact on the NAT module, even when the NAT-MIB module is not implemented at the middlebox.

The impact of MIDCOM rules on the NAT resources is important because a MIDCOM rule not only can create BINDs and NAT sessions, but also is capable of modifying the NAT objects that already exist. For example, FlowDirection and MaxIdleTime parameters in a MIDCOM rule directly affect the TranslationEntity and MaxIdleTime of the associated NAT bind object. Likewise, MaxIdleTime in a MIDCOM rule

has a direct impact on the MaxIdleTime of the associated NAT session object. The lifetime parameter in the MIDCOM rule directly impacts the lifetime of all the impacted NAT BIND and NAT session objects.

8.2. Monitoring Firewall Resources

When a MIDCOM-MIB policy rule is established at a middlebox with firewall capabilities, this may lead to the creation of one or more new firewall rules. Note that in general a single firewall rule per MIDCOM-MIB policy rule will be sufficient. For each policy rule, a MIDCOM client can explore the corresponding firewall filter rule by reading the midcomResourceEntry in the midcomResourceTable that corresponds to the midcomRuleEntry describing the rule. The identification of the firewall filter rule is stored in object midcomRscFirewallRuleId. The value of midcomRscFirewallRuleId may correspond directly to any firewall filter rule number or to an entry in a locally available firewall MIB module.

9. Definitions

The following MIB module imports from [RFC2578], [RFC2579], [RFC2580], [RFC2863], [RFC3411], [RFC4001], and [RFC4008].

```
MIDCOM-MIB DEFINITIONS ::= BEGIN

IMPORTS
    MODULE-IDENTITY, OBJECT-TYPE,
    NOTIFICATION-TYPE, Unsigned32,
    Counter32, Gauge32, mib-2
        FROM SNMPv2-SMI                 -- RFC 2578

    TEXTUAL-CONVENTION, TruthValue,
    StorageType, RowStatus
        FROM SNMPv2-TC                  -- RFC 2579

    MODULE-COMPLIANCE, OBJECT-GROUP,
    NOTIFICATION-GROUP
        FROM SNMPv2-CONF                -- RFC 2580

    SnmpAdminString
        FROM SNMP-FRAMEWORK-MIB         -- RFC 3411

    InetAddressType, InetAddress,
    InetPortNumber,
    InetAddressPrefixLength
        FROM INET-ADDRESS-MIB           -- RFC 4001
```

```
        InterfaceIndexOrZero
            FROM IF-MIB                         -- RFC 2863

        NatBindIdOrZero
            FROM NAT-MIB;                       -- RFC 4008

    midcomMIB MODULE-IDENTITY
        LAST-UPDATED "200708091011Z"  -- August 09, 2007
        ORGANIZATION "IETF Middlebox Communication Working Group"
        CONTACT-INFO
           "WG charter:
              http://www.ietf.org/html.charters/midcom-charter.html

            Mailing Lists:
              General Discussion: midcom@ietf.org
              To Subscribe: midcom-request@ietf.org
              In Body: subscribe your_email_address

            Co-editor:
              Juergen Quittek
              NEC Europe Ltd.
              Kurfuersten-Anlage 36
              69115 Heidelberg
              Germany
              Tel: +49 6221 4342-115
              Email: quittek@nw.neclab.eu

            Co-editor:
              Martin Stiemerling
              NEC Europe Ltd.
              Kurfuersten-Anlage 36
              69115 Heidelberg
              Germany
              Tel: +49 6221 4342-113
              Email: stiemerling@nw.neclab.eu

            Co-editor:
              Pyda Srisuresh
              Kazeon Systems, Inc.
              1161 San Antonio Rd.
              Mountain View, CA 94043
              U.S.A.
              Tel: +1 408 836-4773
              Email: srisuresh@yahoo.com"

        DESCRIPTION
            "This MIB module defines a set of basic objects for
             configuring middleboxes, such as firewalls and network
```

```
            address translators, in order to enable communication
            across these devices.

            Managed objects defined in this MIB module are structured
            in three kinds of objects:
              - transaction objects required according to the MIDCOM
                protocol requirements defined in RFC 3304 and according
                to the MIDCOM protocol semantics defined in RFC 3989,
              - configuration objects that can be used for retrieving or
                setting parameters of the implementation of transaction
                objects,
              - optional monitoring objects that provide information
                about used resource and statistics

            The transaction objects are organized in two subtrees:
              - objects modeling MIDCOM policy rules in the
                midcomRuleTable
              - objects modeling MIDCOM policy rule groups in the
                midcomGroupTable

            Note that typically, configuration objects are not intended
            to be written by MIDCOM clients.  In general, write access
            to these objects needs to be restricted more strictly than
            write access to objects in the transaction subtrees.

            Copyright (C) The Internet Society (2008).  This version
            of this MIB module is part of RFC 5190;  see the RFC
            itself for full legal notices."

        REVISION    "200708091011Z"  -- August 09, 2007
        DESCRIPTION "Initial version, published as RFC 5190."
        ::= { mib-2 171 }

    --
    -- main components of this MIB module
    --

    midcomNotifications   OBJECT IDENTIFIER ::= { midcomMIB 0 }
    midcomObjects         OBJECT IDENTIFIER ::= { midcomMIB 1 }
    midcomConformance     OBJECT IDENTIFIER ::= { midcomMIB 2 }

    --  Transaction objects required according to the MIDCOM
    --  protocol requirements defined in RFC 3304 and according to
    --  the MIDCOM protocol semantics defined in RFC 3989
    midcomTransaction     OBJECT IDENTIFIER ::= { midcomObjects 1 }

    --  Configuration objects that can be used for retrieving
    --  middlebox capability information (mandatory) and for
```

```
   --  setting parameters of the implementation of transaction
   --  objects (optional)
   midcomConfig   OBJECT IDENTIFIER ::= { midcomObjects 2 }

   --  Optional monitoring objects that provide information about
   --  used resource and statistics
   midcomMonitoring      OBJECT IDENTIFIER ::= { midcomObjects 3 }

   --
   -- Transaction Objects
   --
   -- Transaction objects are structured according to the MIDCOM
   -- protocol semantics into two groups:
   --   - objects modeling MIDCOM policy rules in the midcomRuleTable
   --   - objects modeling MIDCOM policy rule groups in the
   --     midcomGroupTable

   --
   -- Policy rule subtree
   --
   -- The midcomRuleTable lists policy rules
   -- including policy reserve rules and policy enable rules.
   --

   midcomRuleTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF MidcomRuleEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "This table lists policy rules.

            It is indexed by the midcomRuleOwner, the
            midcomGroupIndex, and the midcomRuleIndex.
            This implies that a rule is a member of exactly
            one group and that group membership cannot
            be changed.

            Entries can be deleted by writing to
            midcomGroupLifetime or midcomRuleLifetime
            and potentially also to midcomRuleStorageTime."
       ::= { midcomTransaction 3 }

   midcomRuleEntry OBJECT-TYPE
       SYNTAX      MidcomRuleEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
           "An entry describing a particular MIDCOM policy rule."
```

```
        INDEX { midcomRuleOwner, midcomGroupIndex, midcomRuleIndex }
        ::= { midcomRuleTable 1 }

    MidcomRuleEntry ::= SEQUENCE {
        midcomRuleOwner                     SnmpAdminString,
        midcomRuleIndex                     Unsigned32,
        midcomRuleAdminStatus               INTEGER,
        midcomRuleOperStatus                INTEGER,
        midcomRuleStorageType               StorageType,
        midcomRuleStorageTime               Unsigned32,
        midcomRuleError                     SnmpAdminString,
        midcomRuleInterface                 InterfaceIndexOrZero,
        midcomRuleFlowDirection             INTEGER,
        midcomRuleMaxIdleTime               Unsigned32,
        midcomRuleTransportProtocol         Unsigned32,
        midcomRulePortRange                 INTEGER,
        midcomRuleInternalIpVersion         InetAddressType,
        midcomRuleExternalIpVersion         InetAddressType,
        midcomRuleInternalIpAddr            InetAddress,
        midcomRuleInternalIpPrefixLength    InetAddressPrefixLength,
        midcomRuleInternalPort              InetPortNumber,
        midcomRuleExternalIpAddr            InetAddress,
        midcomRuleExternalIpPrefixLength    InetAddressPrefixLength,
        midcomRuleExternalPort              InetPortNumber,
        midcomRuleInsideIpAddr              InetAddress,
        midcomRuleInsidePort                InetPortNumber,
        midcomRuleOutsideIpAddr             InetAddress,
        midcomRuleOutsidePort               InetPortNumber,
        midcomRuleLifetime                  Unsigned32,
        midcomRuleRowStatus                 RowStatus
    }

    midcomRuleOwner OBJECT-TYPE
        SYNTAX      SnmpAdminString (SIZE (0..32))
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The manager who owns this row in the midcomRuleTable.

             This object SHOULD uniquely identify an authenticated
             MIDCOM client.  This object is part of the table index to
             allow for the use of the SNMPv3 View-based Access Control
             Model (VACM, RFC 3415)."
        ::= { midcomRuleEntry 1 }

    midcomRuleIndex OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  not-accessible
```

```
        STATUS      current
        DESCRIPTION
            "The value of this object must be unique in
             combination with the values of the objects
             midcomRuleOwner and midcomGroupIndex in this row."
        ::= { midcomRuleEntry 3 }

    midcomRuleAdminStatus OBJECT-TYPE
        SYNTAX      INTEGER {
                        reserve(1),
                        enable(2),
                        notSet(3)
                    }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The value of this object indicates the desired status of
             the policy rule.  See the definition of midcomRuleOperStatus
             for a description of the values.

             When a midcomRuleEntry is created without explicitly setting
             this object, its value will be notSet(3).

             However, a SET request can only set this object to either
             reserve(1) or enable(2).  Attempts to set this object to
             notSet(3) will always fail with an 'inconsistentValue'
             error.  Note that this error code is SNMP specific.  If the
             MIB module is used with other protocols than SNMP, errors
             with similar semantics specific to those protocols should
             be returned.

             When the midcomRuleAdminStatus object is set, then the
             MIDCOM-MIB implementation will try to read the respective
             relevant objects of the entry and try to achieve the
             corresponding midcomRuleOperStatus.

             Setting midcomRuleAdminStatus to value reserve(1) when
             object midcomRuleOperStatus has a value of reserved(7)
             does not have any effect on the policy rule.
             Setting midcomRuleAdminStatus to value enable(2) when
             object midcomRuleOperStatus has a value of enabled(8)
             does not have any effect on the policy rule.

             Depending on whether the midcomRuleAdminStatus is set to
             reserve(1) or enable(2), several objects must be set in
             advance.  They serve as parameters of the policy rule to be
             established.
```

```
            When object midcomRuleAdminStatus is set to reserve(1),
            then the following objects in the same entry are of
            relevance:
                - midcomRuleInterface
                - midcomRuleTransportProtocol
                - midcomRulePortRange
                - midcomRuleInternalIpVersion
                - midcomRuleExternalIpVersion
                - midcomRuleInternalIpAddr
                - midcomRuleInternalIpPrefixLength
                - midcomRuleInternalPort
                - midcomRuleLifetime

            MIDCOM-MIB implementation may also consider the value
            of object midcomRuleMaxIdleTime when establishing
            a reserve rule.

            When object midcomRuleAdminStatus is set to enable(2),
            then the following objects in the same entry are of
            relevance:
                - midcomRuleInterface
                - midcomRuleFlowDirection
                - midcomRuleMaxIdleTime
                - midcomRuleTransportProtocol
                - midcomRulePortRange
                - midcomRuleInternalIpVersion
                - midcomRuleExternalIpVersion
                - midcomRuleInternalIpAddr
                - midcomRuleInternalIpPrefixLength
                - midcomRuleInternalPort
                - midcomRuleExternalIpAddr
                - midcomRuleExternalIpPrefixLength
                - midcomRuleExternalPort
                - midcomRuleLifetime

            When retrieved, the object returns the last set value.
            If no value has been set, it returns the default value
            notSet(3)."
        DEFVAL { notSet }
        ::= { midcomRuleEntry 4 }

    midcomRuleOperStatus OBJECT-TYPE
        SYNTAX      INTEGER {
                        newEntry(1),
                        setting(2),
                        checkingRequest(3),
                        incorrectRequest(4),
                        processingRequest(5),
```

```
                    requestRejected(6),
                    reserved(7),
                    enabled(8),
                    timedOut(9),
                    terminatedOnRequest(10),
                    terminated(11),
                    genericError(12)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "The actual status of the policy rule.  The
         midcomRuleOperStatus object may have the following values:

         - newEntry(1) indicates that the entry in the
           midcomRuleTable was created, but not modified yet.
           Such an entry needs to be filled with values specifying
           a request first.

         - setting(2) indicates that the entry has been already
           modified after generating it, but no request was made
           yet.

         - checkingRequest(3) indicates that midcomRuleAdminStatus
           has recently been set and that the MIDCOM-MIB
           implementation is currently checking the parameters of
           the request.  This is a transient state.  The value of
           this object will change to either incorrectRequest(4)
           or processingRequest(5) without any external
           interaction.  A MIDCOM-MIB implementation MAY return
           this value while checking request parameters.

         - incorrectRequest(4) indicates that checking a request
           resulted in detecting an incorrect value in one of the
           objects containing request parameters.  The failure
           reason is indicated by the value of midcomRuleError.

         - processingRequest(5) indicates that
           midcomRuleAdminStatus has recently been set and that
           the MIDCOM-MIB implementation is currently processing
           the request and trying to configure the middlebox
           accordingly.  This is a transient state.  The value of
           this object will change to either requestRejected(6),
           reserved(7), or enabled(8) without any external
           interaction.  A MIDCOM-MIB implementation MAY return
           this value while processing a request.

         - requestRejected(6) indicates that a request to establish
```

a policy rule specified by the entry was rejected. The reason for rejection is indicated by the value of midcomRuleError.

- reserved(7) indicates that the entry describes an established policy reserve rule.
  These values of MidcomRuleEntry are meaningful for a reserved policy rule:
    - midcomRuleMaxIdleTime
    - midcomRuleInterface
    - midcomRuleTransportProtocol
    - midcomRulePortRange
    - midcomRuleInternalIpVersion
    - midcomRuleExternalIpVersion
    - midcomRuleInternalIpAddr
    - midcomRuleInternalIpPrefixLength
    - midcomRuleInternalPort
    - midcomRuleOutsideIpAddr
    - midcomRuleOutsidePort
    - midcomRuleLifetime

- enabled(8) indicates that the entry describes an established policy enable rule.
  These values of MidcomRuleEntry are meaningful for an enabled policy rule:

    - midcomRuleFlowDirection
    - midcomRuleInterface
    - midcomRuleMaxIdleTime
    - midcomRuleTransportProtocol
    - midcomRulePortRange
    - midcomRuleInternalIpVersion
    - midcomRuleExternalIpVersion
    - midcomRuleInternalIpAddr
    - midcomRuleInternalIpPrefixLength
    - midcomRuleInternalPort
    - midcomRuleExternalIpAddr
    - midcomRuleExternalIpPrefixLength
    - midcomRuleExternalPort
    - midcomRuleInsideIpAddr
    - midcomRuleInsidePort
    - midcomRuleOutsideIpAddr
    - midcomRuleOutsidePort
    - midcomRuleLifetime

- timedOut(9) indicates that the lifetime of a previously established policy rule has expired and that the policy rule is terminated for this reason.

```
            - terminatedOnRequest(10) indicates that a previously
              established policy rule was terminated by an SNMP
              manager setting the midcomRuleLifetime to 0 or
              setting midcomGroupLifetime to 0.

            - terminated(11) indicates that a previously established
              policy rule was terminated by the MIDCOM-MIB
              implementation for a reason other than lifetime
              expiration or an explicit request from a MIDCOM client.

            - genericError(12) indicates that the policy rule
              specified by the entry is not established due to
              an error condition not listed above.

            The states timedOut(9), terminatedOnRequest(10), and
            terminated(11) are referred to as termination states.

            The states incorrectRequest(4), requestRejected(6),
            and genericError(12) are referred to as error states.

            The checkingRequest(3) and processingRequest(5)
            states are transient states, which will lead to either
            one of the error states or the reserved(7) state or the
            enabled(8) state.  MIDCOM-MIB implementations MAY return
            these values when checking or processing requests."
        DEFVAL { newEntry }
        ::= { midcomRuleEntry 5 }

    midcomRuleStorageType OBJECT-TYPE
        SYNTAX      StorageType
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "When retrieved, this object returns the storage
             type of the policy rule.  Writing to this object can
             change the storage type of the particular row from
             volatile(2) to nonVolatile(3) or vice versa.

             Attempts to set this object to permanent will always
             fail with an 'inconsistentValue' error.  Note that this
             error code is SNMP specific.  If the MIB module is used
             with other protocols than SNMP, errors with similar
             semantics specific to those protocols should be
             returned.

             If midcomRuleStorageType has the value permanent(4),
             then all objects in this row whose MAX-ACCESS value
             is read-create must be read-only."
```

```
        DEFVAL { volatile }
        ::= { midcomRuleEntry 6 }

    midcomRuleStorageTime OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The value of this object specifies how long this row
             can exist in the midcomRuleTable after the
             midcomRuleOperStatus switched to a termination state or
             to an error state.  This object returns the remaining
             time that the row may exist before it is aged out.

             After expiration or termination of the context, the value
             of this object ticks backwards.  The entry in the
             midcomRuleTable is destroyed when the value reaches 0.

             The value of this object may be set in order to increase
             or reduce the remaining time that the row may exist.
             Setting the value to 0 will destroy this entry as soon as
             the midcomRuleOperStatus switched to a termination state
             or to an error state.

             Note that there is no guarantee that the row is stored as
             long as this object indicates.  At any time, the MIDCOM-
             MIB implementation may decide to remove a row describing
             a terminated policy rule before the storage time of the
             corresponding row in the midcomRuleTable reaches the
             value of 0.  In this case, the information stored in this
             row is not available anymore.

             If object midcomRuleStorageType indicates that the policy
             rule has the storage type permanent(4), then this object has
             a constant value of 4294967295."
        DEFVAL { 0 }
        ::= { midcomRuleEntry 7 }

    midcomRuleError OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "This object contains a descriptive error message if
             the transition into the operational status reserved(7)
             or enabled(8) failed.  Implementations must reset the
             error message to a zero-length string when a new
```

```
            attempt to change the policy rule status to reserved(7)
            or enabled(8) is started.

            RECOMMENDED values to be returned in particular cases
            include
              - 'lack of IP addresses'
              - 'lack of port numbers'
              - 'lack of resources'
              - 'specified NAT interface does not exist'
              - 'specified NAT interface does not support NAT'
              - 'conflict with already existing policy rule'
              - 'no internal IP wildcarding allowed'
              - 'no external IP wildcarding allowed'

            The semantics of these error messages and the corresponding
            behavior of the MIDCOM-MIB implementation are specified
            in sections 2.3.9 and 2.3.10 of RFC 3989."
        REFERENCE
            "RFC 3989, sections 2.3.9 and 2.3.10"
        DEFVAL { ''H }
        ::= { midcomRuleEntry 8 }

    midcomRuleInterface OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This object indicates the IP interface for which
            enforcement of a policy rule is requested or performed,
            respectively.

            The interface is identified by its index in the ifTable
            (see IF-MIB in RFC 2863).  If the object has a value of 0,
            then no particular interface is indicated.

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.

            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1) or setting(2), then this object can be
            written by a manager in order to request its preference
            concerning the interface at which it requests NAT service.
            The default value of 0 indicates that the manager does not
            have a preferred interface or does not have sufficient
            topology information for specifying one.  Writing to this
            object in any state other than newEntry(1) or setting(2)
            will always fail with an 'inconsistentValue' error.
```

```
            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value reserved(7) or enabled(8), then this object indicates
            the interface at which NAT service for this rule is
            performed.  If NAT service is not required for enforcing
            the policy rule, then the value of this object is 0.  Also,
            if the MIDCOM-MIB implementation cannot indicate an
            interface, because it does not have this information or
            because NAT service is not offered at a particular single
            interface, then the value of the object is 0.

            Note that the index of a particular interface in the
            ifTable may change after a re-initialization of the
            middlebox, for example, after adding another interface to
            it.  In such a case, the value of this object may change,
            but the interface referred to by the MIDCOM-MIB MUST still
            be the same.  If, after a re-initialization of the
            middlebox, the interface referred to before
            re-initialization cannot be uniquely mapped anymore to a
            particular entry in the ifTable, then the value of object
            midcomRuleOperStatus of the same entry MUST be changed to
            terminated(11).

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 0 }
        ::= { midcomRuleEntry 9 }

    midcomRuleFlowDirection OBJECT-TYPE
        SYNTAX      INTEGER {
                        inbound(1),
                        outbound(2),
                        biDirectional(3)
                    }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "This parameter specifies the direction of enabled
             communication, either inbound(1), outbound(2), or
             biDirectional(3).

             The semantics of this object depends on the protocol
             the rule relates to.  If the rule is independent of
```

the transport protocol (midcomRuleTransportProtocol has a value of 0) or if the transport protocol is UDP, then the value of midcomRuleFlowDirection indicates the direction of packets traversing the middlebox.

In this case, value inbound(1) indicates that packets are traversing from outside to inside, value outbound(2) indicates that packets are traversing from inside to outside. For both values, inbound(1) and outbound(2) packets can traverse the middlebox only unidirectional. A bidirectional flow is indicated by value biDirectional(3).

If the transport protocol is TCP, the packet flow is always bidirectional, but the value of midcomRuleFlowDirection indicates that:

- inbound(1): bidirectional TCP packet flow. First packet, with TCP SYN flag set, must arrive at an outside interface of the middlebox.

- outbound(2): bidirectional TCP packet flow. First packet, with TCP SYN flag set, must arrive at an inside interface of the middlebox.

- biDirectional(3): bidirectional TCP packet flow. First packet, with TCP SYN flag set, may arrive at an inside or an outside interface of the middlebox.

This object is used as input to a request for establishing a policy enable rule as well as for indicating the properties of an established policy rule.

If object midcomRuleOperStatus of the same entry has a value of either newEntry(1), setting(2), or reserved(7), then this object can be written by a manager in order to specify a requested direction to be enabled by a policy rule. Writing to this object in any state other than newEntry(1), setting(2), or reserved(7) will always fail with an 'inconsistentValue' error.

Note that this error code is SNMP specific. If the MIB module is used with other protocols than SNMP, errors with similar semantics specific to those protocols should be returned.

If object midcomRuleOperStatus of the same entry has the value enabled(8), then this object indicates the enabled

```
            flow direction.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { outbound }
        ::= { midcomRuleEntry 10 }

    midcomRuleMaxIdleTime OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "Maximum idle time of the policy rule in seconds.

            If no packet to which the policy rule applies passes the
            middlebox for the specified midcomRuleMaxIdleTime, then
            the policy rule enters the termination state timedOut(9).

            A value of 0 indicates that the policy does not require
            an individual idle time and that instead, a default idle
            time chosen by the middlebox is used.

            A value of 4294967295 ( = 2^32 - 1 ) indicates that the
            policy does not time out if it is idle.

            This object is used as input to a request for
            establishing a policy enable rule as well as for
            indicating the properties of an established policy rule.

            If object midcomRuleOperStatus of the same entry has a
            value of either newEntry(1), setting(2), or reserved(7),
            then this object can be written by a manager in order to
            specify a maximum idle time for the policy rule to be
            requested.  Writing to this object in any state others
            than newEntry(1), setting(2), or reserved(7) will always
            fail with an 'inconsistentValue' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value enabled(8), then this object indicates the maximum
            idle time of the policy rule.  Note that even if a maximum
            idle time greater than zero was requested, the middlebox
```

```
            may not be able to support maximum idle times and set the
            value of this object to zero when entering state
            enabled(8).

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 0 }
        ::= { midcomRuleEntry 11 }

    midcomRuleTransportProtocol OBJECT-TYPE
        SYNTAX      Unsigned32 (0..255)
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The transport protocol.

            Valid values for midcomRuleTransportProtocol
            other than zero are defined at:
            http://www.iana.org/assignments/protocol-numbers

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.

            If object midcomRuleOperStatus of the same entry has a
            value of either newEntry(1) or setting(2), then this
            object can be written by a manager in order to specify a
            requested transport protocol.  If translation of an IP
            address only is requested, then this object must have the
            default value 0.  Writing to this object in any state
            other than newEntry(1) or setting(2) will always fail
            with an 'inconsistentValue' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value reserved(7) or enabled(8), then this object
            indicates which transport protocol is enforced by this
            policy rule.  A value of 0 indicates a rule acting on IP
            addresses only.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
```

```
        DEFVAL { 0 }
        ::= { midcomRuleEntry 12 }

    midcomRulePortRange OBJECT-TYPE
        SYNTAX      INTEGER {
                        single(1),
                        pair(2)
                    }
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The range of port numbers.

             This object is used as input to a request for establishing
             a policy rule as well as for indicating the properties of
             an established policy rule.  It is relevant to the
             operation of the MIDCOM-MIB implementation only if the
             value of object midcomTransportProtocol in the same entry
             has a value other than 0.

             If object midcomRuleOperStatus of the same entry has the
             value newEntry(1) or setting(2), then this object can be
             written by a manager in order to specify the requested
             size of the port range.  With single(1) just a single
             port number is requested, with pair(2) a consecutive pair
             of port numbers is requested with the lower number being
             even.  Requesting a consecutive pair of port numbers may
             be used by RTP [RFC3550] and may even be required to
             support older RTP applications.

             Writing to this object in any state other than
             newEntry(1), setting(2) or reserved(7) will always fail
             with an 'inconsistentValue' error.

             Note that this error code is SNMP specific.  If the MIB
             module is used with other protocols than SNMP, errors with
             similar semantics specific to those protocols should be
             returned.

             If object midcomRuleOperStatus of the same entry has a
             value of either reserved(7) or enabled(8), then this
             object will have the value that it had before the
             transition to this state.

             If object midcomRuleOperStatus of the same entry has a
             value other than newEntry(1), setting(2), reserved(7), or
             enabled(8), then the value of this object is irrelevant."
        DEFVAL { single }
```

```
        ::= { midcomRuleEntry 13}

    midcomRuleInternalIpVersion OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "IP version of the internal address (A0) and the inside
             address (A1).  Allowed values are ipv4(1), ipv6(2),
             ipv4z(3), and ipv6z(4).

             This object is used as input to a request for establishing
             a policy rule as well as for indicating the properties of
             an established policy rule.

             If object midcomRuleOperStatus of the same entry has the
             value newEntry(1) or setting(2), then this object can be
             written by a manager in order to specify the IP version
             required at the inside of the middlebox.  Writing to this
             object in any state other than newEntry(1) or setting(2)
             will always fail with an 'inconsistentValue' error.

             Note that this error code is SNMP specific.  If the MIB
             module is used with other protocols than SNMP, errors with
             similar semantics specific to those protocols should be
             returned.

             If object midcomRuleOperStatus of the same entry has the
             value reserved(7) or enabled(8), then this object
             indicates the internal/inside IP version.

             If object midcomRuleOperStatus of the same entry has a
             value other than newEntry(1), setting(2), reserved(7), or
             enabled(8), then the value of this object is irrelevant."
        DEFVAL { ipv4 }
        ::= { midcomRuleEntry 14 }

    midcomRuleExternalIpVersion OBJECT-TYPE
        SYNTAX      InetAddressType
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "IP version of the external address (A3) and the outside
             address (A2).  Allowed values are ipv4(1) and ipv6(2).

             This object is used as input to a request for establishing
             a policy rule as well as for indicating the properties of
             an established policy rule.
```

```
            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1) or setting(2), then this object can be
            written by a manager in order to specify the IP version
            required at the outside of the middlebox.  Writing to
            this object in any state other than newEntry(1) or
            setting(2) will always fail with an 'inconsistentValue'
            error.
            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value reserved(7) or enabled(8), then this object
            indicates the external/outside IP version.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7) or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { ipv4 }
        ::= { midcomRuleEntry 15 }

    midcomRuleInternalIpAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The internal IP address (A0).

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.

            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1) or setting(2), then this object can be
            written by a manager in order to specify the internal IP
            address for which a reserve policy rule or a enable policy
            rule is requested to be established.  Writing to this
            object in any state other than newEntry(1) or setting(2)
            will always fail with an 'inconsistentValue' error.
            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value reserved(7) or enabled(8), then this object will
            have the value which it had before the transition to this
```

```
            state.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7) or
            enabled(8), then the value of this object is irrelevant."
        ::= { midcomRuleEntry 16 }

    midcomRuleInternalIpPrefixLength OBJECT-TYPE
        SYNTAX      InetAddressPrefixLength
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The prefix length of the internal IP address used for
             wildcarding.  A value of 0 indicates a full wildcard;
             in this case, the value of midcomRuleInternalIpAddr is
             irrelevant.  If midcomRuleInternalIpVersion has a value
             of ipv4(1), then a value > 31 indicates no wildcarding
             at all.  If midcomRuleInternalIpVersion has a value
             of ipv4(2), then a value > 127 indicates no wildcarding
             at all.  A MIDCOM-MIB implementation that does not
             support IP address wildcarding MUST implement this object
             as read-only with a value of 128.  A MIDCOM that does
             not support wildcarding based on prefix length MAY
             restrict allowed values for this object to 0 and 128.

             This object is used as input to a request for establishing
             a policy rule as well as for indicating the properties of
             an established policy rule.

             If object midcomRuleOperStatus of the same entry has the
             value newEntry(1) or setting(2), then this object can be
             written by a manager in order to specify the prefix length
             of the internal IP address for which a reserve policy rule
             or an enable policy rule is requested to be established.
             Writing to this object in any state other than newEntry(1)
             or setting(2) will always fail with an 'inconsistentValue'
             error.

             Note that this error code is SNMP specific.  If the MIB
             module is used with other protocols than SNMP, errors with
             similar semantics specific to those protocols should be
             returned.

             If object midcomRuleOperStatus of the same entry has the
             value reserved(7) or enabled(8), then this object will
             have the value which it had before the transition to this
             state.
```

```
            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 128 }
        ::= { midcomRuleEntry 17 }

    midcomRuleInternalPort OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The internal port number.  A value of 0 is a wildcard.

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.  It is relevant to the
            operation of the MIDCOM-MIB implementation only if the
            value of object midcomTransportProtocol in the same entry
            has a value other than 0.

            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1) or setting(2), then this object can be
            written by a manager in order to specify the internal port
            number for which a reserve policy rule or an enable policy
            rule is requested to be established.  Writing to this
            object in any state other than newEntry(1) or setting(2)
            will always fail with an 'inconsistentValue' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value reserved(7) or enabled(8), then this object will
            have the value that it had before the transition to this
            state.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 0 }
        ::= { midcomRuleEntry 18 }

    midcomRuleExternalIpAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-create
        STATUS      current
```

```
        DESCRIPTION
            "The external IP address (A3).

             This object is used as input to a request for establishing
             a policy rule as well as for indicating the properties of
             an established policy rule.

             If object midcomRuleOperStatus of the same entry has the
             value newEntry(1), setting(2), or reserved(7), then this
             object can be written by a manager in order to specify the
             external IP address for which an enable policy rule is
             requested to be established.  Writing to this object in
             any state other than newEntry(1), setting(2), or reserved(7)
             will always fail with an 'inconsistentValue' error.

             Note that this error code is SNMP specific.  If the MIB
             module is used with other protocols than SNMP, errors with
             similar semantics specific to those protocols should be
             returned.

             If object midcomRuleOperStatus of the same entry has the
             value enabled(8), then this object will have the value
             that it had before the transition to this state.

             If object midcomRuleOperStatus of the same entry has a
             value other than newEntry(1), setting(2), reserved(7), or
             enabled(8), then the value of this object is irrelevant."
        ::= { midcomRuleEntry 19 }

    midcomRuleExternalIpPrefixLength OBJECT-TYPE
        SYNTAX      InetAddressPrefixLength
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The prefix length of the external IP address used for
             wildcarding.  A value of 0 indicates a full wildcard;
             in this case, the value of midcomRuleExternalIpAddr is
             irrelevant.  If midcomRuleExternalIpVersion has a value
             of ipv4(1), then a value > 31 indicates no wildcarding
             at all.  If midcomRuleExternalIpVersion has a value
             of ipv4(2), then a value > 127 indicates no wildcarding
             at all.  A MIDCOM-MIB implementation that does not
             support IP address wildcarding MUST implement this object
             as read-only with a value of 128.  A MIDCOM that does
             not support wildcarding based on prefix length MAY
             restrict allowed values for this object to 0 and 128.

             This object is used as input to a request for establishing
```

```
            a policy rule as well as for indicating the properties of
            an established policy rule.

            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1), setting(2), or reserved(7), then this
            object can be written by a manager in order to specify the
            prefix length of the external IP address for which an
            enable policy rule is requested to be established.
            Writing to this object in any state other than
            newEntry(1), setting(2), or reserved(7) will always fail
            with an 'inconsistentValue' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value enabled(8), then this object will have the value
            that it had before the transition to this state.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 128 }
        ::= { midcomRuleEntry 20 }

    midcomRuleExternalPort OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The external port number.  A value of 0 is a wildcard.

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.  It is relevant to the
            operation of the MIDCOM-MIB implementation only if the
            value of object midcomTransportProtocol in the same entry
            has a value other than 0.

            If object midcomRuleOperStatus of the same entry has the
            value newEntry(1), setting(2) or reserved(7), then this
            object can be written by a manager in order to specify the
            external port number for which an enable policy rule is
            requested to be established.  Writing to this object in
            any state other than newEntry(1), setting(2) or reserved(7)
            will always fail with an 'inconsistentValue' error.
```

```
            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has the
            value enabled(8), then this object will have the value
            which it had before the transition to this state.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7) or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 0 }
        ::= { midcomRuleEntry 21 }

    midcomRuleInsideIpAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The inside IP address at the middlebox (A1).

            The value of this object is relevant only if
            object midcomRuleOperStatus of the same entry has
            a value of either reserved(7) or enabled(8)."
        ::= { midcomRuleEntry 22 }

    midcomRuleInsidePort OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The inside port number at the middlebox.
             A value of 0 is a wildcard.

            The value of this object is relevant only if
            object midcomRuleOperStatus of the same entry has
            a value of either reserved(7) or enabled(8)."
        ::= { midcomRuleEntry 23 }

    midcomRuleOutsideIpAddr OBJECT-TYPE
        SYNTAX      InetAddress
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The outside IP address at the middlebox (A2).

            The value of this object is relevant only if
```

```
            object midcomRuleOperStatus of the same entry has
            a value of either reserved(7) or enabled(8)."
        ::= { midcomRuleEntry 24 }

    midcomRuleOutsidePort OBJECT-TYPE
        SYNTAX      InetPortNumber
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
            "The outside port number at the middlebox.
            A value of 0 is a wildcard.

            The value of this object is relevant only if
            object midcomRuleOperStatus of the same entry has
            a value of either reserved(7) or enabled(8)."
        ::= { midcomRuleEntry 25 }

    midcomRuleLifetime OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "The remaining lifetime in seconds of this policy rule.

            Lifetime of a policy rule starts when object
            midcomRuleOperStatus in the same entry enters either
            state reserved(7) or state enabled(8).

            This object is used as input to a request for establishing
            a policy rule as well as for indicating the properties of
            an established policy rule.

            If object midcomRuleOperStatus of the same entry has a
            value of either newEntry(1) or setting(2), then this
            object can be written by a manager in order to specify
            the requested lifetime of a policy rule to be established.

            If object midcomRuleOperStatus of the same entry has a
            value of either reserved(7) or enabled(8), then this
            object indicates the (continuously decreasing) remaining
            lifetime of the established policy rule.  Note that when
            entering state reserved(7) or enabled(8), the MIDCOM-MIB
            implementation can choose a lifetime shorter than the one
            requested.

            Unlike other parameters of the policy rule, this parameter
            can still be written in state reserved(7) and enabled(8).
```

```
            Writing to this object is processed by the MIDCOM-MIB
            implementation by choosing a lifetime value that is
            greater than 0 and less than or equal to the minimum of
            the requested value and the value specified by object
            midcomConfigMaxLifetime:

             0 <= lt_granted <= MINIMUM(lt_requested, lt_maximum)

            where:
               - lt_granted is the actually granted lifetime by the
                 MIDCOM-MIB implementation
               - lt_requested is the requested lifetime of the MIDCOM
                 client
               - lt_maximum is the value of object
                 midcomConfigMaxLifetime

            SNMP SET requests to this object may be rejected or the
            value of the object after an accepted SET operation may be
            less than the value that was contained in the SNMP SET
            request.

            Successfully writing a value of 0 terminates the policy
            rule.  Note that after a policy rule is terminated, still
            the entry will exist as long as indicated by the value of
            midcomRuleStorageTime.

            Writing to this object in any state other than
            newEntry(1), setting(2), reserved(7), or enabled(7)
            will always fail with an 'inconsistentValue' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            If object midcomRuleOperStatus of the same entry has a
            value other than newEntry(1), setting(2), reserved(7), or
            enabled(8), then the value of this object is irrelevant."
        DEFVAL { 180 }
        ::= { midcomRuleEntry 26 }

    midcomRuleRowStatus OBJECT-TYPE
        SYNTAX      RowStatus
        MAX-ACCESS  read-create
        STATUS      current
        DESCRIPTION
            "A control that allows entries to be added and removed from
            this table.
```

```
            Entries can also be removed from this table by setting
            objects midcomRuleLifetime and midcomRuleStorageTime of
            an entry to 0.

            Attempts to set a row notInService(2) where the value
            of the midcomRuleStorageType object is permanent(4) or
            readOnly(5) will result in an 'notWritable' error.

            Note that this error code is SNMP specific.  If the MIB
            module is used with other protocols than SNMP, errors with
            similar semantics specific to those protocols should be
            returned.

            The value of this object has no effect on whether other
            objects in this conceptual row can be modified."
        ::= { midcomRuleEntry 27 }

    --
    -- Policy rule group subtree
    --
    -- The midcomGroupTable lists all current policy rule groups.
    --

    midcomGroupTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF MidcomGroupEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table lists all current policy rule groups.

            Entries in this table are created or removed
            implicitly when entries in the midcomRuleTable are
            created or removed, respectively.  A group entry
            in this table only exists as long as there are
            member rules of this group in the midcomRuleTable.

            The table serves for listing the existing groups and
            their remaining lifetimes and for changing lifetimes
            of groups and implicitly of all group members.
            Groups and all their member policy rules can only be
            deleted by deleting all member policies in the
            midcomRuleTable.

            Setting midcomGroupLifetime will result in setting
            the lifetime of all policy members to the same value."
        ::= { midcomTransaction 4 }

    midcomGroupEntry OBJECT-TYPE
```

```
        SYNTAX      MidcomGroupEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An entry describing properties of a particular
             MIDCOM policy rule group."
        INDEX { midcomRuleOwner, midcomGroupIndex }
        ::= { midcomGroupTable 1 }

    MidcomGroupEntry ::= SEQUENCE {
        midcomGroupIndex      Unsigned32,
        midcomGroupLifetime   Unsigned32
    }

    midcomGroupIndex OBJECT-TYPE
        SYNTAX      Unsigned32 (1..4294967295)
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of this group for the midcomRuleOwner.
             A group is identified by the combination of
             midcomRuleOwner and midcomGroupIndex.

             The value of this index must be unique per
             midcomRuleOwner."
        ::= { midcomGroupEntry 2 }

    midcomGroupLifetime OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "When retrieved, this object delivers the maximum
             lifetime in seconds of all member rules of this group,
             i.e., of all rows in the midcomRuleTable that have the
             same values for midcomRuleOwner and midcomGroupIndex.

             Successfully writing to this object modifies the
             lifetime of all member policies.  Successfully
             writing a value of 0 terminates all member policies
             and implicitly deletes the group as soon as all member
             entries are removed from the midcomRuleTable.

             Note that after a group's lifetime is expired or is
             set to 0, still the corresponding entry in the
             midcomGroupTable will exist as long as terminated
             member policy rules are stored as entries in the
```

```
            midcomRuleTable.

            Writing to this object is processed by the MIDCOM-MIB
            implementation by choosing a lifetime value that is
            greater than 0 and less than or equal to the minimum of
            the requested value and the value specified by object
            midcomConfigMaxLifetime:

             0 <= lt_granted <= MINIMUM(lt_requested, lt_maximum)

            where:
               - lt_granted is the actually granted lifetime by the
                 MIDCOM-MIB implementation
               - lt_requested is the requested lifetime of the MIDCOM
                 client
               - lt_maximum is the value of object
                 midcomConfigMaxLifetime

            SNMP SET requests to this object may be rejected or the
            value of the object after an accepted SET operation may be
            less than the value that was contained in the SNMP SET
            request."
        ::= { midcomGroupEntry 3 }

    --
    -- Configuration Objects
    --
    --  Configuration objects that can be used for retrieving
    --  middlebox capability information (mandatory) and for
    --  setting parameters of the implementation of transaction
    --  objects (optional).
    --
    --  Note that typically configuration objects are not intended
    --  to be written by MIDCOM clients.  In general, write access
    --  to these objects needs to be restricted more strictly than
    --  write access to transaction objects.
    --

    --
    -- Capabilities subtree
    --
    -- This subtree contains objects to which MIDCOM clients should
    -- have read access.
    --

    midcomConfigMaxLifetime OBJECT-TYPE
        SYNTAX      Unsigned32
        UNITS       "seconds"
```

```
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "When retrieved, this object returns the maximum lifetime,
             in seconds, that this middlebox allows policy rules to
             have."
        ::= { midcomConfig 1 }

    midcomConfigPersistentRules OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "When retrieved, this object returns true(1) if the
             MIDCOM-MIB implementation can store policy rules
             persistently.  Otherwise, it returns false(2).

             A value of true(1) indicates that there may be
             entries in the midcomRuleTable with object
             midcomRuleStorageType set to value nonVolatile(3)."
        ::= { midcomConfig 2 }

    midcomConfigIfTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF MidcomConfigIfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "This table indicates capabilities of the MIDCOM-MIB
             implementation per IP interface.

             The table is indexed by the object midcomConfigIfIndex.

             For indexing a single interface, this object contains
             the value of the ifIndex object that is associated
             with the interface.  If an entry with
             midcomConfigIfIndex = 0 occurs, then bits set in
             objects of this entry apply to all interfaces for which
             there is no entry in this table with the interface's
             index."
        ::= { midcomConfig 3 }

    midcomConfigIfEntry OBJECT-TYPE
        SYNTAX      MidcomConfigIfEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "An entry describing the capabilities of a middlebox
             with respect to the indexed IP interface."
```

```
        INDEX { midcomConfigIfIndex }
        ::= { midcomConfigIfTable 1 }

    MidcomConfigIfEntry ::= SEQUENCE {
        midcomConfigIfIndex          InterfaceIndexOrZero,
        midcomConfigIfBits           BITS,
        midcomConfigIfEnabled        TruthValue
    }

    midcomConfigIfIndex OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of an entry in the midcomConfigIfTable.

             For values different from zero, this object
             identifies an IP interface by containing the same
             value as the ifIndex object associated with the
             interface.

             Note that the index of a particular interface in the
             ifTable may change after a re-initialization of the
             middlebox, for example, after adding another interface to
             it.  In such a case, the value of this object may change,
             but the interface referred to by the MIDCOM-MIB MUST still
             be the same.  If, after a re-initialization of the
             middlebox, the interface referred to before
             re-initialization cannot be uniquely mapped anymore to a
             particular entry in the ifTable, then the value of object
             midcomConfigIfEnabled of the same entry MUST be changed to
             false(2).

             If the object has a value of 0, then values
             specified by further objects of the same entry
             apply to all interfaces for which there is no
             explicit entry in the midcomConfigIfTable."
        ::= { midcomConfigIfEntry 1 }

    midcomConfigIfBits OBJECT-TYPE
        SYNTAX      BITS {
                        ipv4(0),
                        ipv6(1),
                        addressWildcards(2),
                        portWildcards(3),
                        firewall(4),
                        nat(5),
                        portTranslation(6),
```

```
                    protocolTranslation(7),
                    twiceNat(8),
                    inside(9)
                }
    MAX-ACCESS  read-only
    STATUS      current
    DESCRIPTION
        "When retrieved, this object returns a set of bits
         indicating the capabilities (or configuration) of
         the middlebox with respect to the referenced IP interface.
         If the index equals 0, then all set bits apply to all
         interfaces.

         If the ipv4(0) bit is set, then the middlebox supports
         IPv4 at the indexed IP interface.

         If the ipv6(1) bit is set, then the middlebox supports
         IPv6 at the indexed IP interface.

         If the addressWildcards(2) bit is set, then the
         middlebox supports IP address wildcarding at the indexed
         IP interface.

         If the portWildcards(3) bit is set, then the
         middlebox supports port wildcarding at the indexed
         IP interface.

         If the firewall(4) bit is set, then the middlebox offers
         firewall functionality at the indexed interface.

         If the nat(5) bit is set, then the middlebox offers
         network address translation service at the indexed
         interface.

         If the portTranslation(6) bit is set, then the middlebox
         offers port translation service at the indexed interface.
         This bit is only relevant if nat(5) is set.

         If the protocolTranslation(7) bit is set, then the
         middlebox offers protocol translation service between
         IPv4 and IPv6 at the indexed interface.  This bit is only
         relevant if nat(5) is set.

         If the twiceNat(8) bit is set, then the middlebox offers
         twice network address translation service at the indexed
         interface.  This bit is only relevant if nat(5) is set.

         If the inside(9) bit is set, then the indexed interface is
```

```
            an inside interface with respect to NAT functionality.
            Otherwise, it is an outside interface.  This bit is only
            relevant if nat(5) is set.  An SNMP agent supporting both
            the MIDCOM-MIB module and the NAT-MIB module SHOULD ensure
            that the value of this object is consistent with the values
            of corresponding objects in the NAT-MIB module."
        ::= { midcomConfigIfEntry 2 }

    midcomConfigIfEnabled OBJECT-TYPE
        SYNTAX      TruthValue
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
            "The value of this object indicates the availability of
            the middlebox service described by midcomConfigIfBits
            at the indexed IP interface.

            By writing to this object, the MIDCOM support for the
            entire IP interface can be switched on or off.  Setting
            this object to false(2) immediately stops middlebox
            support at the indexed IP interface.  This implies that
            all policy rules that use NAT or firewall resources at
            the indexed IP interface are terminated immediately.
            In this case, the MIDCOM agent MUST send
            midcomUnsolicitedRuleEvent to all MIDCOM clients that
            have access to one of the terminated rules."
        DEFVAL { true }
        ::= { midcomConfigIfEntry 3 }

    --
    -- Firewall subtree
    --
    -- This subtree contains the firewall configuration table
    --

    midcomConfigFirewallTable OBJECT-TYPE
        SYNTAX      SEQUENCE OF MidcomConfigFirewallEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "This table lists the firewall configuration per IP interface.

            It can be used for configuring how policy rules created by
            MIDCOM clients are realized as firewall rules of a firewall
            implementation.  Particularly, the priority used for MIDCOM
            policy rules can be configured.  For a single firewall
            implementation at a particular IP interface, all MIDCOM
            policy rules are realized as firewall rules with the same
```

```
            priority.  Also, a firewall rule group name can be
            configured.

            The table is indexed by the object midcomConfigFirewallIndex.
            For indexing a single interface, this object contains the
            value of the ifIndex object that is associated with the
            interface.  If an entry with midcomConfigFirewallIndex = 0
            occurs, then bits set in objects of this entry apply to all
            interfaces for which there is no entry in this table for the
            interface's index."
        ::= { midcomConfig 4 }

    midcomConfigFirewallEntry OBJECT-TYPE
        SYNTAX      MidcomConfigFirewallEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "An entry describing a particular set of
            firewall resources."
        INDEX { midcomConfigFirewallIndex }
        ::= { midcomConfigFirewallTable 1 }

    MidcomConfigFirewallEntry ::= SEQUENCE {
        midcomConfigFirewallIndex      InterfaceIndexOrZero,
        midcomConfigFirewallGroupId    SnmpAdminString,
        midcomConfigFirewallPriority   Unsigned32
    }

    midcomConfigFirewallIndex OBJECT-TYPE
        SYNTAX      InterfaceIndexOrZero
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
            "The index of an entry in the midcomConfigFirewallTable.

             For values different from 0, this object identifies an
             IP interface by containing the same value as the ifIndex
             object associated with the interface.

             Note that the index of a particular interface in the
             ifTable may change after a re-initialization of the
             middlebox, for example, after adding another interface to
             it.  In such a case, the value of this object may change,
             but the interface referred to by the MIDCOM-MIB MUST still
             be the same.  If, after a re-initialization of the
             middlebox, the interface referred to before
             re-initialization cannot be uniquely mapped anymore to a
             particular entry in the ifTable, then the entry in the
```

```
            midcomConfigFirewallTable MUST be deleted.

            If the object has a value of 0, then values specified by
            further objects of the same entry apply to all interfaces
            for which there is no explicit entry in the
            midcomConfigFirewallTable."
        ::= { midcomConfigFirewallEntry 1 }

    midcomConfigFirewallGroupId OBJECT-TYPE
        SYNTAX      SnmpAdminString
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
           "The firewall rule group to which all firewall rules are
            assigned that the MIDCOM server creates for the interface
            indicated by object midcomConfigFirewallIndex.  If the
            value of object midcomConfigFirewallIndex is 0, then all
            firewall rules of the MIDCOM server that are created for
            interfaces with no specific entry in the
            midcomConfigFirewallTable are assigned to the firewall
            rule group indicated by the value of this object."
        ::= { midcomConfigFirewallEntry 2 }

    midcomConfigFirewallPriority OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-write
        STATUS      current
        DESCRIPTION
           "The priority assigned to all firewall rules that the
            MIDCOM server creates for the interface indicated by
            object midcomConfigFirewallIndex.  If the value of object
            midcomConfigFirewallIndex is 0, then this priority is
            assigned to all firewall rules of the MIDCOM server that
            are created for interfaces for which there is no specific
            entry in the midcomConfigFirewallTable."
        ::= { midcomConfigFirewallEntry 3 }

    --
    -- Monitoring Objects
    --
    -- Monitoring objects are structured into two groups,
    -- the midcomResourceGroup providing information about used
    -- resources and the midcomStatisticsGroup providing information
    -- about MIDCOM transaction statistics.

    --
    -- Resources subtree
    --
```

```
   -- The MIDCOM resources subtree contains a set of managed
   -- objects describing the currently used resources of NAT
   -- and firewall implementations.
   --

   --
   -- Textual conventions for objects of the resource subtree
   --

   MidcomNatBindMode ::= TEXTUAL-CONVENTION
       STATUS      current
       DESCRIPTION
          "An indicator of the kind of NAT resources used by a policy
           rule.  This definition corresponds to the definition of
           NatBindMode in the NAT-MIB (RFC 4008).  Value none(3) can
           be used to indicate that the policy rule does not use
           any NAT binding.
           "
       SYNTAX      INTEGER {
                       addressBind(1),
                       addressPortBind(2),
                       none(3)
                   }

   MidcomNatSessionIdOrZero ::= TEXTUAL-CONVENTION
       DISPLAY-HINT "d"
       STATUS      current
       DESCRIPTION
          "A unique ID that is assigned to each NAT session by
           a NAT implementation.  This definition corresponds to
           the definition of NatSessionId in the NAT-MIB (RFC 4008).
           Value 0 can be used to indicate that the policy rule does
           not use any NAT binding."
       SYNTAX      Unsigned32

   --
   -- The MIDCOM resource table
   --

   midcomResourceTable OBJECT-TYPE
       SYNTAX      SEQUENCE OF MidcomResourceEntry
       MAX-ACCESS  not-accessible
       STATUS      current
       DESCRIPTION
          "This table lists all used middlebox resources per
           MIDCOM policy rule.

           The midcomResourceTable augments the
```

```
            midcomRuleTable."
        ::= { midcomMonitoring 1 }

    midcomResourceEntry OBJECT-TYPE
        SYNTAX      MidcomResourceEntry
        MAX-ACCESS  not-accessible
        STATUS      current
        DESCRIPTION
           "An entry describing a particular set of middlebox
            resources."
        AUGMENTS { midcomRuleEntry }
        ::= { midcomResourceTable 1 }

    MidcomResourceEntry ::= SEQUENCE {
        midcomRscNatInternalAddrBindMode   MidcomNatBindMode,
        midcomRscNatInternalAddrBindId     NatBindIdOrZero,
        midcomRscNatInsideAddrBindMode     MidcomNatBindMode,
        midcomRscNatInsideAddrBindId       NatBindIdOrZero,
        midcomRscNatSessionId1             MidcomNatSessionIdOrZero,
        midcomRscNatSessionId2             MidcomNatSessionIdOrZero,
        midcomRscFirewallRuleId            Unsigned32
    }

    midcomRscNatInternalAddrBindMode OBJECT-TYPE
        SYNTAX      MidcomNatBindMode
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "An indication of whether this policy rule uses an address
            NAT bind or an address-port NAT bind for binding the
            internal address.

            If the MIDCOM-MIB module is operated together with
            the NAT-MIB module (RFC 4008) then object
            midcomRscNatInternalAddrBindMode contains the same
            value as the corresponding object
            natSessionPrivateSrcEPBindMode of the NAT-MIB module."
        ::= { midcomResourceEntry 4 }

    midcomRscNatInternalAddrBindId OBJECT-TYPE
        SYNTAX      NatBindIdOrZero
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "This object references to the allocated internal NAT
            bind that is used by this policy rule.  A NAT bind
            describes the mapping of internal addresses to
            outside addresses.  MIDCOM-MIB implementations can
```

```
            read this object to learn the corresponding NAT bind
            resource for this particular policy rule.

            If the MIDCOM-MIB module is operated together with
            the NAT-MIB module (RFC 4008) then object
            midcomRscNatInternalAddrBindId contains the same
            value as the corresponding object
            natSessionPrivateSrcEPBindId of the NAT-MIB module."
        ::= { midcomResourceEntry 5 }

    midcomRscNatInsideAddrBindMode OBJECT-TYPE
        SYNTAX      MidcomNatBindMode
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "An indication of whether this policy rule uses an address
            NAT bind or an address-port NAT bind for binding the
            external address.

            If the MIDCOM-MIB module is operated together with
            the NAT-MIB module (RFC 4008), then object
            midcomRscNatInsideAddrBindMode contains the same
            value as the corresponding object
            natSessionPrivateDstEPBindMode of the NAT-MIB module."
        ::= { midcomResourceEntry 6 }

    midcomRscNatInsideAddrBindId OBJECT-TYPE
        SYNTAX      NatBindIdOrZero
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "This object refers to the allocated external NAT
            bind that is used by this policy rule.  A NAT bind
            describes the mapping of external addresses to
            inside addresses.  MIDCOM-MIB implementations can
            read this object to learn the corresponding NAT bind
            resource for this particular policy rule.

            If the MIDCOM-MIB module is operated together with the
            NAT-MIB module (RFC 4008), then object
            midcomRscNatInsideAddrBindId contains the same
            value as the corresponding object
            natSessionPrivateDstEPBindId of the NAT-MIB module."
        ::= { midcomResourceEntry 7 }

    midcomRscNatSessionId1 OBJECT-TYPE
        SYNTAX      MidcomNatSessionIdOrZero
        MAX-ACCESS  read-only
```

```
        STATUS      current
        DESCRIPTION
           "This object refers to the first allocated NAT session for
            this policy rule.  MIDCOM-MIB implementations can read this
            object to learn whether or not a NAT session for a
            particular policy rule is used.  A value of 0 means that no
            NAT session is allocated for this policy rule.  A value
            other than 0 refers to the NAT session."
       ::= { midcomResourceEntry 8 }

    midcomRscNatSessionId2 OBJECT-TYPE
        SYNTAX      MidcomNatSessionIdOrZero
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "This object refers to the second allocated NAT session for
            this policy rule.  MIDCOM-MIB implementations can read this
            object to learn whether or not a NAT session for a
            particular policy rule is used.  A value of 0 means that no
            NAT session is allocated for this policy rule.  A value
            other than 0 refers to the NAT session."
        ::= { midcomResourceEntry 9 }

    midcomRscFirewallRuleId OBJECT-TYPE
        SYNTAX      Unsigned32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "This object refers to the allocated firewall
            rule in the firewall engine for this policy rule.
            MIDCOM-MIB implementations can read this value to
            learn whether a firewall rule for this particular
            policy rule is used or not.  A value of 0 means that
            no firewall rule is allocated for this policy rule.
            A value other than 0 refers to the firewall rule
            number within the firewall engine."
        ::= { midcomResourceEntry 10 }

    --
    -- Statistics subtree
    --
    -- The MIDCOM statistics subtree contains a set of managed
    -- objects providing statistics about the usage of transaction
    -- objects.
    --

    midcomStatistics      OBJECT IDENTIFIER ::= { midcomMonitoring 2 }
```

```
   midcomCurrentOwners OBJECT-TYPE
       SYNTAX      Gauge32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The number of different values for midcomRuleOwner
           for all current entries in the midcomRuleTable."
       ::= { midcomStatistics 1 }

   midcomTotalRejectedRuleEntries OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of failed attempts to create an entry
           in the midcomRuleTable."
       ::= { midcomStatistics 2 }

   midcomCurrentRulesIncomplete OBJECT-TYPE
       SYNTAX      Gauge32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The current number of policy rules that are incomplete.

           Policy rules are loaded via row entries in the
           midcomRuleTable.  This object counts policy rules that are
           loaded but not fully specified, i.e., they are in state
           newEntry(1) or setting(2)."
       ::= { midcomStatistics 3 }

   midcomTotalIncorrectReserveRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of policy reserve rules that failed
           parameter check and entered state incorrectRequest(4)."
       ::= { midcomStatistics 4 }

   midcomTotalRejectedReserveRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of policy reserve rules that failed
           while being processed and entered state requestRejected(6)."
       ::= { midcomStatistics 5 }
```

```
   midcomCurrentActiveReserveRules OBJECT-TYPE
       SYNTAX      Gauge32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The number of currently active policy reserve rules."
       ::= { midcomStatistics 6 }

   midcomTotalExpiredReserveRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of expired policy reserve rules
           (entered termination state timedOut(9))."
       ::= { midcomStatistics 7 }

   midcomTotalTerminatedOnRqReserveRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of policy reserve rules that were
           terminated on request (entered termination state
           terminatedOnRequest(10))."
       ::= { midcomStatistics 8 }

   midcomTotalTerminatedReserveRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of policy reserve rules that were
           terminated, but not on request (entered termination state
           terminated(11))."
       ::= { midcomStatistics 9 }

   midcomTotalIncorrectEnableRules OBJECT-TYPE
       SYNTAX      Counter32
       MAX-ACCESS  read-only
       STATUS      current
       DESCRIPTION
          "The total number of policy enable rules that failed
           parameter check and entered state incorrectRequest(4)."
       ::= { midcomStatistics 10 }

   midcomTotalRejectedEnableRules OBJECT-TYPE
       SYNTAX      Counter32
```

```
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The total number of policy enable rules that failed
            while being processed and entered state requestRejected(6)."
        ::= { midcomStatistics 11 }
    midcomCurrentActiveEnableRules OBJECT-TYPE
        SYNTAX      Gauge32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The number of currently active policy enable rules."
        ::= { midcomStatistics 12 }

    midcomTotalExpiredEnableRules OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The total number of expired policy enable rules
            (entered termination state timedOut(9))."
        ::= { midcomStatistics 13 }

    midcomTotalTerminatedOnRqEnableRules OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The total number of policy enable rules that were
            terminated on request (entered termination state
            terminatedOnRequest(10))."
        ::= { midcomStatistics 14 }

    midcomTotalTerminatedEnableRules OBJECT-TYPE
        SYNTAX      Counter32
        MAX-ACCESS  read-only
        STATUS      current
        DESCRIPTION
           "The total number of policy enable rules that were
            terminated, but not on request (entered termination state
            terminated(11))."
        ::= { midcomStatistics 15 }

    --
    -- Notifications.
    --

    midcomUnsolicitedRuleEvent NOTIFICATION-TYPE
```

```
        OBJECTS     { midcomRuleOperStatus, midcomRuleLifetime }
        STATUS      current
        DESCRIPTION
            "This notification is generated whenever the value of
             midcomRuleOperStatus enters any error state or any
             termination state without an explicit trigger by a
             MIDCOM client."
        ::= { midcomNotifications 1 }

    midcomSolicitedRuleEvent NOTIFICATION-TYPE
        OBJECTS     { midcomRuleOperStatus, midcomRuleLifetime }
        STATUS      current
        DESCRIPTION
            "This notification is generated whenever the value
             of midcomRuleOperStatus enters one of the states
             {reserved, enabled, any error state, any termination state}
             as a result of a MIDCOM agent writing successfully to
             object midcomRuleAdminStatus.

             In addition, it is generated when the lifetime of
             a rule was changed by successfully writing to object
             midcomRuleLifetime."
        ::= { midcomNotifications 2 }

    midcomSolicitedGroupEvent NOTIFICATION-TYPE
        OBJECTS     { midcomGroupLifetime }
        STATUS      current
        DESCRIPTION
            "This notification is generated for indicating that the
             lifetime of all member rules of the group was changed by
             successfully writing to object midcomGroupLifetime.

             Note that this notification is only sent if the lifetime
             of a group was changed by successfully writing to object
             midcomGroupLifetime.  No notification is sent
               - if a group's lifetime is changed by writing to object
                 midcomRuleLifetime of any of its member policies,
               - if a group's lifetime expires (in this case,
                 notifications are sent for all member policies), or
               - if the group is terminated by terminating the last
                 of its member policies without writing to object
                 midcomGroupLifetime."
        ::= { midcomNotifications 3 }

    --
    -- Conformance information
    --
```

```
   midcomCompliances OBJECT IDENTIFIER ::= { midcomConformance 1 }
   midcomGroups      OBJECT IDENTIFIER ::= { midcomConformance 2 }

   --
   -- compliance statements
   --

   -- This is the MIDCOM compliance definition ...

   --

   midcomCompliance MODULE-COMPLIANCE
       STATUS      current
       DESCRIPTION
           "The compliance statement for implementations of the
            MIDCOM-MIB module.

            Note that compliance with this compliance
            statement requires compliance with the
            ifCompliance3 MODULE-COMPLIANCE statement of the
            IF-MIB [RFC2863]."
       MODULE      -- this module
       MANDATORY-GROUPS {
               midcomRuleGroup,
               midcomNotificationsGroup,
               midcomCapabilitiesGroup,
               midcomStatisticsGroup
       }
       GROUP   midcomConfigFirewallGroup
       DESCRIPTION
          "A compliant implementation does not have to implement
           the midcomConfigFirewallGroup."
       GROUP   midcomResourceGroup
       DESCRIPTION
          "A compliant implementation does not have to implement
           the midcomResourceGroup."
       OBJECT midcomRuleInternalIpPrefixLength
       MIN-ACCESS  read-only
       DESCRIPTION
          "Write access is not required.  When write access is
           not supported, return 128 as the value of this object.
           A value of 128 means that the function represented by
           this option is not supported."
       OBJECT midcomRuleExternalIpPrefixLength
       MIN-ACCESS  read-only
       DESCRIPTION
          "Write access is not required.  When write access is
           not supported, return 128 as the value of this object.
```

```
            A value of 128 means that the function represented by
            this option is not supported."
        OBJECT midcomRuleMaxIdleTime
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required.  When write access is
            not supported, return 0 as the value of this object.
            A value of 0 means that the function represented by
            this option is not supported."
        OBJECT midcomRuleInterface
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        OBJECT midcomConfigMaxLifetime
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        OBJECT midcomConfigPersistentRules
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        OBJECT midcomConfigIfEnabled
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        OBJECT midcomConfigFirewallGroupId
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        OBJECT midcomConfigFirewallPriority
        MIN-ACCESS  read-only
        DESCRIPTION
           "Write access is not required."
        ::= { midcomCompliances 1 }

    midcomRuleGroup OBJECT-GROUP
        OBJECTS {
            midcomRuleAdminStatus,
            midcomRuleOperStatus,
            midcomRuleStorageType,
            midcomRuleStorageTime,
            midcomRuleError,
            midcomRuleInterface,
            midcomRuleFlowDirection,
            midcomRuleMaxIdleTime,
            midcomRuleTransportProtocol,
            midcomRulePortRange,
            midcomRuleInternalIpVersion,
```

```
            midcomRuleExternalIpVersion,
            midcomRuleInternalIpAddr,
            midcomRuleInternalIpPrefixLength,
            midcomRuleInternalPort,
            midcomRuleExternalIpAddr,
            midcomRuleExternalIpPrefixLength,
            midcomRuleExternalPort,
            midcomRuleInsideIpAddr,
            midcomRuleInsidePort,
            midcomRuleOutsideIpAddr,
            midcomRuleOutsidePort,
            midcomRuleLifetime,
            midcomRuleRowStatus,
            midcomGroupLifetime
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information about
             policy rules and policy rule groups."
        ::= { midcomGroups 1 }

    midcomCapabilitiesGroup OBJECT-GROUP
        OBJECTS {
            midcomConfigMaxLifetime,
            midcomConfigPersistentRules,
            midcomConfigIfBits,
            midcomConfigIfEnabled
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information about
             the capabilities of a middlebox."
        ::= { midcomGroups 2 }

    midcomConfigFirewallGroup OBJECT-GROUP
        OBJECTS {
            midcomConfigFirewallGroupId,
            midcomConfigFirewallPriority
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information about
             the firewall rule group and firewall rule priority to
             be used by firewalls loaded through MIDCOM."
        ::= { midcomGroups 3 }

    midcomResourceGroup OBJECT-GROUP
        OBJECTS {
```

```
            midcomRscNatInternalAddrBindMode,
            midcomRscNatInternalAddrBindId,
            midcomRscNatInsideAddrBindMode,
            midcomRscNatInsideAddrBindId,
            midcomRscNatSessionId1,
            midcomRscNatSessionId2,
            midcomRscFirewallRuleId
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing information about
             the used NAT and firewall resources."
        ::= { midcomGroups 4 }

    midcomStatisticsGroup OBJECT-GROUP
        OBJECTS {
            midcomCurrentOwners,
            midcomTotalRejectedRuleEntries,
            midcomCurrentRulesIncomplete,
            midcomTotalIncorrectReserveRules,
            midcomTotalRejectedReserveRules,
            midcomCurrentActiveReserveRules,
            midcomTotalExpiredReserveRules,
            midcomTotalTerminatedOnRqReserveRules,
            midcomTotalTerminatedReserveRules,
            midcomTotalIncorrectEnableRules,
            midcomTotalRejectedEnableRules,
            midcomCurrentActiveEnableRules,
            midcomTotalExpiredEnableRules,
            midcomTotalTerminatedOnRqEnableRules,
            midcomTotalTerminatedEnableRules
        }
        STATUS      current
        DESCRIPTION
            "A collection of objects providing statistical
             information about the MIDCOM server."
        ::= { midcomGroups 5 }
```

```
   midcomNotificationsGroup NOTIFICATION-GROUP
        NOTIFICATIONS {
            midcomUnsolicitedRuleEvent,
            midcomSolicitedRuleEvent,
            midcomSolicitedGroupEvent
        }
        STATUS    current
        DESCRIPTION
            "The notifications emitted by the midcomMIB."
        ::= { midcomGroups 6 }

   END
```

10. Security Considerations

Obviously, securing access to firewall and NAT configuration is extremely important for maintaining network security. This section first describes general security issues of the MIDCOM-MIB module and then discusses three concrete security threats: unauthorized middlebox configuration, unauthorized access to middlebox configuration information, and unauthorized access to the MIDCOM service configuration.

10.1. General Security Issues

There are a number of management objects defined in this MIB module with a MAX-ACCESS clause of read-write and/or read-create. Such objects may be considered sensitive or vulnerable in some network environments. But also access to managed objects with a MAX-ACCESS clause of read-only may be considered sensitive or vulnerable. The support for SET and GET operations in a non-secure environment without proper protection can have a negative effect on network operations.

SNMP versions prior to SNMPv3 did not include adequate security. Even if the network itself is secure (for example by using IPsec), even then, there is no control as to who on the secure network is allowed to access and GET/SET (read/change/create/delete) the objects in this MIB module.

Deployment of SNMP versions prior to SNMPv3 is NOT RECOMMENDED.

Compliant MIDCOM-MIB implementations MUST support SNMPv3 security services including data integrity, identity authentication, data confidentiality, and replay protection.

It is REQUIRED that the implementations support the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 3414 [RFC3414] and the View-based Access Control Model RFC 3415 [RFC3415] is RECOMMENDED.

It is then a customer/operator responsibility to ensure that the SNMP entity giving access to an instance of this MIB is properly configured to give access to the objects only to those principals (users) that have legitimate rights to indeed GET or SET (change/create/delete) them.

To facilitate the provisioning of access control by a security administrator using the View-based Access Control Model (VACM) defined in RFC 3415 [RFC3415] for tables in which multiple users may need to independently create or modify entries, the initial index is used as an "owner index". This is supported by the midcomRuleTable and the midcomGroupTable. Each of them uses midcomRuleOwner as the initial index. midcomRuleOwner has the syntax of SnmpAdminString, and can thus be trivially mapped to an SNMP securityName or a groupName as defined in VACM, in accordance with a security policy.

All entries in the two mentioned tables belonging to a particular user will have the same value for this initial index. For a given user's entries in a particular table, the object identifiers for the information in these entries will have the same subidentifiers (except for the "column" subidentifier) up to the end of the encoded owner index. To configure VACM to permit access to this portion of the table, one would create vacmViewTreeFamilyTable entries with the value of vacmViewTreeFamilySubtree including the owner index portion, and vacmViewTreeFamilyMask "wildcarding" the column subidentifier. More elaborate configurations are possible.

10.2. Unauthorized Middlebox Configuration

The most dangerous threat to network security related to the MIDCOM-MIB module is unauthorized access to facilities for establishing policy rules. In such a case, unauthorized principals would write to the midcomRuleTable for opening firewall pinholes and/or for creating NAT maps, bindings, and/or sessions. Establishing policies can be used to gain access to networks and systems that are protected by firewalls and/or NATs.

If this protection is removed by unauthorized access to MIDCOM-MIB policies, then the resulting degradation of network security can be severe. Confidential information protected by a firewall might become accessible to unauthorized principals, attacks exploiting

security leaks of systems in the protected network might become possible from external networks, and it might be possible to stop firewalls blocking denial-of-service attacks.

MIDCOM-MIB implementations MUST provide means for strict authentication, message integrity check, and write access control to managed objects that can be used for establishing policy rules. These are objects in the midcomRuleTable and midcomGroupTable with a MAX-ACCESS clause of read-write and/or read-create.

Particularly sensitive is write access to the managed object midcomRuleAdminStatus, because writing it causes policy rules to be established.

Also, writing to other managed objects in the two tables can make security vulnerable if it interferes with the authorized establishment of a policy rule, for example, by wildcarding a policy rule after the corresponding entry in the midcomRuleTable is created, but before the authorized owner establishes the rule by writing to midcomRuleAdminStatus.

Not only unauthorized establishment, but also unauthorized lifetime extension of an existing policy rule may be considered sensitive or vulnerable in some network environments. Therefore, means for strict authentication, message integrity check, and write access control to managed object midcomGroupLifetime MUST be provided by MIDCOM-MIB implementations.

10.3. Unauthorized Access to Middlebox Configuration

Another threat to network security is unauthorized access to entries in the midcomRuleTable. The entries contain information about existing pinholes in the firewall and/or about the current NAT configuration. This information can be used for attacking the internal network from outside. Therefore, a MIDCOM-MIB implementation MUST also provide means for read access control to the midcomRuleTable.

Also, a MIDCOM-MIB implementation SHOULD provide means for protecting different authenticated MIDCOM agents from each other, such that, for example, an authenticated user can only read entries in the midcomRuleTable for which the initial index midcomRuleOwner matches the client's SNMP securityName or VACM groupName.

10.4. Unauthorized Access to MIDCOM Service Configuration

There are three objects with a MAX-ACCESS clause of read-write that configure the MIDCOM service: midcomConfigIfEnabled, midcomFirewallGroupId, and midcomFirewallPriority.

Unauthorized writing to object midcomConfigIfEnabled can cause serious interruptions of network service.

Writing to midcomFirewallGroupId and/or midcomFirewallPriority can be used to increase or reduce the priority of firewall rules that are generated when a policy rule is established in the midcomRuleTable. Increasing the priority might permit firewall rules generated via the MIDCOM-MIB module to overrule basic security rules at the firewall that should have higher priority than the ones generated via the MIDCOM-MIB module.

Therefore, also for these objects, means for strict control of write access MUST be provided by a MIDCOM-MIB implementation.

11. Acknowledgements

This memo is based on a long history of discussion within the MIDCOM MIB design team. Many thanks to Mary Barnes, Jeff Case, Wes Hardaker, David Harrington, and Tom Taylor for fruitful comments and recommendations and to Juergen Schoenwaelder acting as a very constructive MIB doctor.

12. IANA Considerations

IANA has assigned an OID for the MIB module in this document:

Descriptor	OBJECT IDENTIFIER value
midcomMIB	{ mib-2 171 }

13. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC5189] Stiemerling, M., Quittek, J., and T. Taylor, "Middlebox Communication (MIDCOM) Protocol Semantics", RFC 5189, March 2008.

[RFC2578] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.

[RFC2579] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC2580] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, June 2000.

[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.

[RFC3413] Levi, D., Meyer, P., and B. Stewart, "Simple Network Management Protocol Applications", STD 62, RFC 3413, December 2002.

[RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.

[RFC3418] Presuhn, R., Ed., "Management Information Base (MIB) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3418, December 2002.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC4001] Daniele, M., Haberman, B., Routhier, S., and J. Schoenwaelder, "Textual Conventions for Internet Network Addresses", RFC 4001, February 2005.

[RFC4008] Rohit, R., Srisuresh, P., Raghunarayan, R., Pai, N., and C. Wang, "Definitions of Managed Objects for Network Address Translators (NAT)", RFC 4008, March 2005.

14. Informative References

[RFC3410] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction and Applicability Statements for Internet-Standard Management Framework", RFC 3410, December 2002.

[RFC3234] Carpenter, B. and S. Brim, "Middleboxes: Taxonomy and Issues", RFC 3234, February 2002.

[RFC3303] Srisuresh, P., Kuthan, J., Rosenberg, J., Molitor, A., and A. Rayhan, "Middlebox communication architecture and framework", RFC 3303, August 2002.

[RFC3304] Swale, R., Mart, P., Sijben, P., Brim, S., and M. Shore, "Middlebox Communications (midcom) Protocol Requirements", RFC 3304, August 2002.

[RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.

Authors' Addresses

Juergen Quittek
NEC Europe Ltd.
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 4342-115
EMail: quittek@nw.neclab.eu

Martin Stiemerling
NEC Europe Ltd.
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 4342-113
EMail: stiemerling@nw.neclab.eu

Pyda Srisuresh
Kazeon Systems, Inc.
1161 San Antonio Rd.
Mountain View, CA 94043
U.S.A.

Phone: +1 408 836 4773
EMail: srisuresh@yahoo.com

Full Copyright Statement

Intellectual Property