Internet Draft
Document: draft-ietf-midcom-mib-analysis-01.txt

M. Barnes
Nortel Networks
Editor

Category: Informational
Expires: April 2004 October 2003

Middlebox Communications (MIDCOM) Protocol Managed Objects Analysis

Status of this Memo

Copyright Notice

Abstract

This document provides an analysis and identification of the managed objects for dynamic configuration of middleboxes. The scope of the middleboxes to which these managed objects apply is limited to NATs and Firewalls. However, the managed objects as identified in this document are intended to provide a baseline for the dynamic configuration of other types of middleboxes. The applicability of existing Management Information Base (MIB) modules to the MIDCOM requirements, framework and semantics is described. Additional managed objects are identified to satisfy the entirety of the MIDCOM requirements, framework and semantics and to provide a complete MIDCOM MIB for NATs and Firewalls to fully realize the requirements of the MIDCOM protocol. The actual definition of any new managed objects is provided in a separate document [MDCMIB].

Table of Contents

Overview

This intent of this document is to provide a detailed analysis of the managed objects for dynamic configuration of middleboxes. The scope of the middleboxes to which these managed objects are specifically applied is limited to NATs and Firewalls. However, the resultant MIB should be extensible and provide the basis for the development of managed objects for configuring other types of middleboxes.

Section 1 provides an overview of the SNMP Management Framework. Section 2 provides further background on SNMP and its applicability to the MIDCOM Protocol Framework, Requirements and semantics.

Section 3 provides a high level overview of some existing MIB modules potentially relevant and reusable, which satisfy the MIDCOM requirements and semantics, and relate to the MIDCOM architecture and framework.

Section 4 provides a detailed discussion of existing MIB modules, defining the level of applicability to the MIDCOM protocol requirements, framework and semantics and re-usability for the MIDCOM MIB.

Section 5 identifies the additional MIDCOM specific managed objects required to satisfy some of the requirements and to provide a linkage between the existing MIB modules applicable to MIDCOM. The actual

definition of the MIB module for new MIDCOM specific managed objects is provided in a separate document [MDCMIB].

Conventions used in this document

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].

1. SNMP Management Framework

For a detailed overview of the documents that describe the current Internet-Standard (SNMP) 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].

2. MIDCOM Overview and SNMP Applicability

The MIDCOM architecture and framework [RFC3303] defines a model in which trusted third parties can be delegated to assist middleboxes in performing their operations, without requiring application intelligence be embedded in the middleboxes. This trusted third party is referred to as the MIDCOM Agent. The MIDCOM protocol is defined between the MIDCOM agent and middlebox.

The SNMP management framework provides functions equivalent to those defined by the MIDCOM framework, although there are a few architectural differences.

For SNMP, application intelligence is captured in MIB modules, rather than in the messaging protocol. MIB modules define a data model of the information that can be collected and configured for managed functionality. The SNMP messaging protocol transports the data in a standardized format without needing to understand the semantics of the data being transferred. The endpoints of the communication understand the semantics of the data.

Traditionally, the SNMP endpoints have been called Manager and Agent. An SNMP manager is an entity capable of generating requests and

receiving notifications, and a SNMP agent is an entity capable of responding to requests and generating notifications. As applied to the MIDCOM framework, the SNMP Manager corresponds to the MIDCOM agent and the SNMP Agent corresponds to the Middlebox.

The MIDCOM protocol is divided into three phases, per section 4 of [RFC3303]:

- Session Setup
- Run-time (involving real-time configuration of the middlebox)
- Session Termination

A MIDCOM session is defined to be a lasting association between a MIDCOM agent and a middlebox. The MIDCOM agent should initiate the session prior to the start of the application. Although the SNMP management framework does not have the concept of a session, session-like associations can be established through the use of managed objects. Requests from the MIDCOM agent to the Middlebox are performed using write access to managed objects defined in MIB modules. The middlebox (SNMP agent) responds to requests by sending an SNMP response message indicating the success or failure of the request. The MIDCOM agent (SNMP manager) MAY verify this information by reading or polling the corresponding managed objects.

The MIDCOM Protocol semantics [MDCSEM] defines two basic transaction types: request transactions and notify transactions. SNMPv3 uses the architecture detailed in [RFC3411], where all SNMP entities are capable of performing certain functions, such as the generation of requests, response to requests, the generation of asynchronous notifications, the receipt of notifications, and the proxy-forwarding of SNMP messages. SNMP is used to read and manipulate a virtual database (the MIB) composed of objects representing commands, controls, status, and statistics, which are defined in managed-application-specific MIB modules.

3. SNMP and the MIDCOM data model

This section provides a high level description and levels of abstraction of the categories of data required to satisfy the MIDCOM requirements and semantics as it relates to existing SNMP MIB modules.

Application-specific MIB modules can be defined at varying levels of abstraction. At the lowest level, vendor-specific, device-specific parameters may be defined, for instance, to configure a specific model of firewall. At a higher level, a MIB module may define an abstracted view of firewall functionality that can be used to specify a firewall policy, which an implementation can translate into the necessary parameters to configure the specific model of firewall on

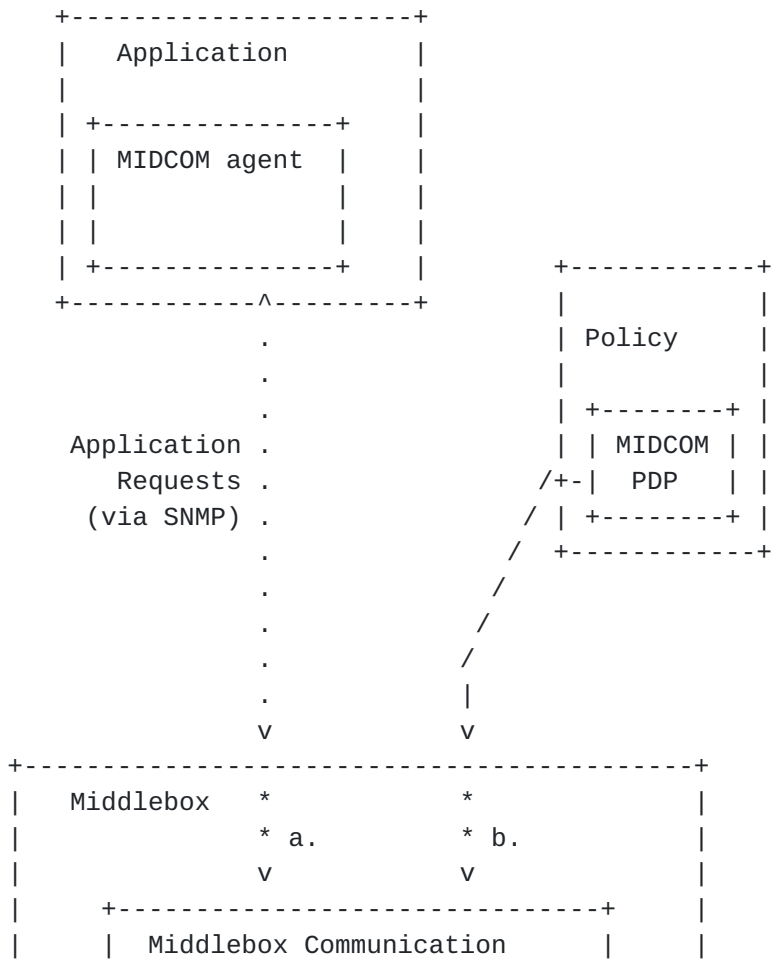
which the abstract MIB is implemented. At a higher level yet, a MIB

module may define service policies or business policies that end up being translated into more detailed instructions, possibly into the more detailed MIB module data schemas. It is common practice to have one MIB module point to other MIB modules that contain less/more concrete conceptual representations.

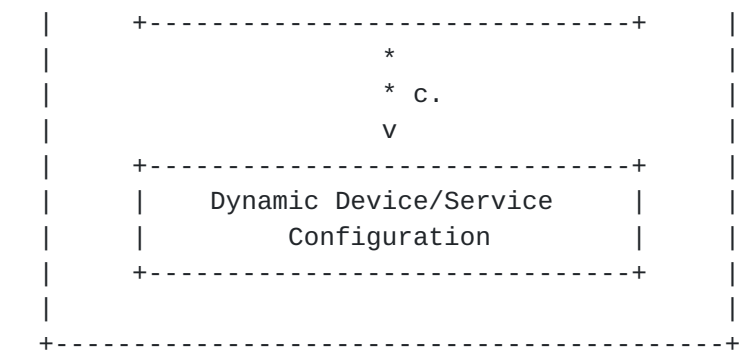
SNMP for the MIDCOM protocol can leverage the data schemas of many existing MIB modules designed to permit secure communications, configuration of devices, configuration of services and policy coordination abstractions. The actual specification of the policies is outside the scope of the MIDCOM protocol.

Many existing MIB modules provide monitoring capabilities that can be applied to MIDCOM functionality.

The following diagram (Figure 1) summarizes the potential relevance and reusability of the data schema of existing MIB models to the MIDCOM architecture to satisfy the MIDCOM protocol framework, requirements and semantics:




| | Protocol (MIDCOM) Interface | |



Legend: .... Middlebox Communication Protocol (MIDCOM)
//// MIDCOM PDP Interface (outside scope of this document)
**** Managed objects relevant to the MIDCOM Interface (with the associated letters referencing the MIB modules potentially applicable summarized below:

a. gaps between existing MIB modules (b and c) and MIDCOM requirements
b. POLICY-BASED-MANAGEMENT-MIB, DIFFSERV-CONFIG-MIB,
c. IPSEC-POLICY-MIB, NAT-MIB, DIFFSERV-MIB

Figure 1: Data relationships relevant to the MIDCOM Interface

3.1 Secure Communications

MIDCOM requirements include mutual authentication, message integrity checking, timeliness checking to prevent replay, message encryption, and authorization controls to ensure only certain agents can modify certain subsets of middlebox configurations. MIDCOM requires secure request-response capabilities and secure notifications.

SNMPv3 is designed to provide secure communications between two end-points. SNMPv3 defines MIB modules to allow the monitoring and configuration of all these security features. They are defined in RFC3411-RFC3418, and RFC3410 provides an overview of these capabilities.

3.2 Device Configuration

SNMP is the most commonly used standardized protocol for remotely monitoring and manipulating the configuration of devices. There are a

large number of IETF standard and vendor-specific MIB modules available.

Most IETF standard MIB modules do not provide much configuration support because SNMPv1 and SNMPv2c were non-secure, and it is difficult to standardize abstractions that provide enough information to configure device implementations that require vendor-specific parameters. There are many vendor-specific MIB modules that permit configuration of the vendor's devices.

SNMP MIB modules are definitions of virtual databases with scalars and tables of data. SNMP supports multiple mechanisms to define relationships between entries in different tables. For example, entries in multiple tables are often related by common indices. SNMP uses a standardized hierarchical namespace, so the value of a field in one table can serve as the index into another table.

The ability to define relationships between MIB module tables (including tables in different MIB modules) allows an abstracted configuration policy to point to a vendor-specific configuration MIB module for more detailed instructions.

There are multiple ways to send policies to middleboxes, including SNMP and COPS/PR and RADIUS/Diameter, and most policies are auto-magically converted into low-level configuration commands that set the correct operational parameters to enforce desired behavior.

Some middlebox functionalities are related to physical and logical topologies that are created by dynamically manipulating device configurations. Some MIB modules that can be used for topology configuration would include the 802.1X MIB [81XMIB] and the Interfaces MIB [RFC2863] to enable or disable a physical port or logical interface, the Bridge MIB [BREMIB]to assign interfaces into virtual LANs and to enable port mirroring functionality for IDS usage, the Layer Two Tunneling MIB or IPSec MIB to create topology tunnels for VPNs, and so on.

There are many IETF standard MIB modules that monitor traffic, which can be used to verify that a policy is being enforced. Most "transmission" MIB modules, those that fall under the { MIB-2 transmission } subtree relative to Interfaces MIB entries, provide statistics about traffic going in or out of ports on a device. The Bridge MIB can be used to monitor the amount of traffic being forwarded into or out of virtual LANs, and so on.

3.3 Service Configuration

A middlebox may be able to support multiple types of services, and a MIDCOM agent must determine which services are available and running, and which have stopped running. Middlebox functionalities are applications that run on a middlebox, and there are multiple MIB modules designed to monitor applications and their operational characteristics. Most of the MIB modules described here are for monitoring only, but could be extended with application-specific MIB modules for configuration and additional monitoring.

The Host Resources MIB [RFC2790] provides monitoring of hardware resources, such as memory and CPU load, and monitors installed applications, running applications, and application performance. These can be used to do capability discovery for a middlebox, and these factors can be important to consider before configuring additional functionality or sessions on a middlebox.

The Network Services Monitoring MIB [RFC2788] module provides objects for monitoring high-level concepts related to network services, such as their current run status and their associations. This MIB works with supplemental service-specific MIB modules, including configuration objects.

The Systems Application MIB [RFC2287] monitors installed applications, running applications, and running processes. The installed application information can be important for determining the actual capabilities of the model and version of firewall installed.

However, MIDCOM is primarily about dynamically configuring middlebox functionality, so MIB modules associated with configuration, specifically any associated with the configuration of firewalls and NATS, are the main focus.

The Diffserv MIB [RFC3289] describes the configuration and management of a Differentiated Services interface in terms of one or more Traffic Conditioning Blocks (TCB), each containing, arranged in the specified order, by definition, zero or more classifiers, meters, actions, algorithmic droppers, queues and schedulers. The "linked-list" approach is very flexible, and could be used to configure some firewall tasks.

The IPSec Policy MIB [IPCMIB] defines objects that could be reused for purposes of filtering service-related traffic and subsequent policy actions.

3.4 Policy Coordination

To properly coordinate policy application, it is necessary to determine if a device has the capabilities needed to effectively enforce a policy, and to coordinate the application of policies according to time constraints, priorities, rule groupings, policy sessions, and so on.

The SNMPCONF working has developed a number of MIB modules designed for the purpose of policy coordination.

Many policies are dependent on factors that are not so much traffic-related as business related. For example, the role that a device serves in the network or the geographic location of a device may impact a policy. The SNMPCONF Policy MIB [PBMMIB] allows an administrator to define roles, and associate them with policies.

The SNMPCONF MIB modules include a policy download table, a policy registration table, and a scheduling function for defining when a policy should be made active and when it should be made dormant. Time schedules can be grouped for easier manipulation, and wildcards are supported. To ease integration with other policy efforts, the schedule table is modeled after the Policy Core Information Model scheduler.

SNMPCONF provides a capabilities table to advertise the functionality available for policy enforcement, including configuration parameters to enable a MIDCOM agent to be notified when new capabilities are installed on a system. Capabilities may be available on some components of a system and not others, such as a board in a chassis, but also may be accessible only in certain logical partitions, such as the community profile (more accurately, the SNMPv3 context) of the super-user.

SNMPCONF defines tracking tables, so an administrator can determine which elements are being controlled by which policies. The MIB also includes debugging tables for logging policy enforcement run-time exceptions. An administrator can disable policies in place, if they desire.

4. Applicability of existing MIB modules

This section summarizes the details of the applicability of existing MIB modules to the MIDCOM data model. As highlighted in Figure 1, the MIDCOM protocol itself is only defined to be the interface from the MIDCOM agent (SNMP manager) to the middlebox or MIDCOM Interface. However, requests from the MIDCOM agent to the MIDCOM Interface must be evaluated against the installed policies and must contain all the data required for the specific device/service configuration. In addition, the session setup reply includes capabilities of the

middlebox, several of which relate to policies. Thus, although the Policy interface itself is out of scope of the MIDCOM protocol, the correlation of the policy related data in the form of rules to the data associated with the MIDCOM Interface is imperative. In effect, an instance of the "MIDCOM MIB" comprises the data from the semantics evaluated against the policy and applied to configure the device/service.

Several of the MIB modules discussed in section 3 were analyzed and and the following were found to have general applicability and varying levels of re-usability for MIDCOM:

- Network Address Translators (NAT) MIB [NATMIB]
- Policy Based Management MIB [PBMMIB]
- IPsec Policy Configuration MIB [IPCMIB]
- Differentiated Services MIB [RFC3289]

4.1 Network Address Translators (NAT) MIB

The NAT MIB module [NATMIB] is intended to be used for configuration as well as monitoring of a device capable of traditional NAT functions. Although, the NAT MIB module appears to meet all of the MIDCOM requirements concerning NAT control, a detailed evaluation against the semantics highlights some areas requiring resolution.

The primary concerns arise from some fundamental views on the MIDCOM MIB and its relation to the NAT (and FW) MIB(s), whether the relationship is explicit or implicit, and whether the MIDCOM Agent has a direct interface to the NAT (or FW) MIB(s).

Taking the perspective that the MIDCOM MIB has an explicit relationship with the NAT MIB and that the MIDCOM Agent has a direct interface to the NAT and FW MIBs results in the following position:

- PRRs have a direct relationship to NAT Binds.
- PERs have a direct relationship to NAT sessions and FW rules.
- Agent specific Group membership IDs should be assignable by agents.

Taking the opposite perspective that the relationship between the MIDCOM MIB and the NAT MIB is implicit and that the MIDCOM agent does not have a direct interface to the NAT MIB and that its interface is abstracted by the MIDCOM MIB results in the following position:

- PRR is an abstract entity, related to binds and address maps.
- PER is an abstract entity whose relationship goes beyond the NAT session.
- Middlebox should assign and manage Group IDs for the agent.

Appendix A was put forth to provide a detailed analysis of some of the specific issues. Once these issues are resolved, the impact will be summarized in this section.

In addition, section 5 summarizes the current MIB proposals the issues requiring resolution and WG consensus.

Additional MIB modules, such as those defined by SNMP Policy Based Management MIB (as described in section 4.2), allowing the definition of policy rulesets and grouping of policy rules are also required.

4.2 Policy Based Management MIB

This MIB defines managed objects that enable policy-based monitoring and management of SNMP infrastructure. The Policy Based Management MIB defines MIB objects for the following areas: roles, capabilities and time.

[Editor's note: Although the policy interface itself to the middlebox is out of scope for the MIDCOM protocol, the rules associated with the MIB module(s) for MIDCOM are in scope and thus it is anticipated that there is some reusability of the managed objects defined by the PBMMIB, rather than of the entire application of this MIB itself. This section will be expanded once more detailed analysis has been completed].

4.3 IPsec Policy Configuration MIB

The IPSEC-POLICY-MIB is a large MIB designed to support IPsec and IKE management in a policy and rule oriented fashion. The MIB module is divided into 3 portions, only one of which would be useful for reuse with the MIDCOM MIB. Specifically, the IPSEC-POLICY-MIB provides a generic mechanism for performing packet processing based on a rule set, anticipated to provide the basis for a general FW MIB. Rules within the IPSEC-POLICY-MIB are generic and simply bind a filter to an action. Filters provided within the IPSEC-POLICY-MIB itself are numerous and fairly complete for most common packet filtering usage but externally defined filters (like those that may need to be developed within a MIDCOM specific MIB module) are supported. The actions encapsulated within the IPSEC-POLICY-MIB are mostly related to IKE and IPsec and thus aren't very useful as applied to MIDCOM. However, actions (like filters) can be externally defined. Compound filter and action sequences can be defined for administrators that need more complex boolean logic or need to chain multiple actions together based on success/failure states. The compound mechanisms are also generic and would let MIDCOM specific MIB elements to be used within the compound bindings if necessary.

[Editor's note: this is an initial analysis; a more detailed analysis to be included once the details are completed. The current proposal is to separate the packet filtering aspects into a separate FW MIB. However, progress on this is pending support for this proposal by the WG involved. Otherwise, an independent FW MIB would need to be defined.]

4.4 Differentiated Services MIB

The Diffserv MIB is a very powerful and flexible MIB module, however, this flexibility is too broad in general for the MIDCOM protocol requirements. In addition, the requirement for NAT support, and specifically policy rule lifetimes in the MIDCOM protocol, further highlight that the Diffserv MIB alone is unsuitable as the MIDCOM MIB Module.

However, the Diffserv model of using different tables for data path elements could be applied to the MIDCOM MIB module. The use of RowPointers as connectors in the Diffserv MIB allows for the simple extension of the MIB. The RowPointers, whether "next" or "specific", may point to Entries defined in other MIB modules. This mechanism can point to other, possibly vendor-specific, configuration MIB modules. In addition, the reuse of some specific definitions out of the DIFFSERV MIB module is worth further consideration for the MIDCOM MIB module, (e.g. the diffServMultiFieldClfrTable).

[Editor's note: Once we start needing to fill in the gaps as highlighted in item a of the diagram in Figure 1, this will be revisited].

4.5 Summary of applicability of existing MIB modules

< To Be Completed >

<Diagram showing these MIB modules as applied to the basic data model>

5. Additional MIDCOM specific managed objects

At this time, there have been two detailed proposals put forth by members of the design team defining the MIDCOM MIB:

- draft-stiemerling-midcom-mib-00.txt
- draft-srisuresh-midcom-mib-00.txt

The primary differences between these two MIBs, as discussed in section 4.1, arise from some fundamental views on the MIDCOM MIB and its relation to the NAT and FW MIBs.

The following summarizes the issues currently being discussed within the design team, some of which are also reflected in the detailed NATMIB analysis in Appendix A and put forth for discussion on the mailing list with regards to the MIDCOM semantics:

1. Session model versus transaction model. There's differing views as to whether MIDCOM is session or transaction oriented. The fundamental issue remains to determine which transactions rely on a session model and how they can be realized through SNMP.

2. Level of abstraction of the MIDCOM MIB from the device. The prevailing view seemed to be that the MIDCOM MIB could be abstracted from the device, however, the flipside of that is that it leaves a lot to implementation choice and thus was deemed a potential pitfall.

3. Interfacing between MIDCOM MIB and existing NAT and FW MIBs. This fundamentally relates to how much integration of MIDCOM should there be with the existing NAT and FW MIBs and whether there are separate MIDCOM "shims" to accomplish this. This relates to the decision on the level of abstraction.

4. Extensions and restrictions to the MIDCOM semantics in support of SNMP specifically (i.e. the semantics were written to be protocol agnostic, however, the selection of SNMP as the MIDCOM protocol imposes the need for some potential extensions and restrictions to the semantics).

A single MIB [MDCMIB] document will be produced once these issues are resolved.

6. Security Considerations

The MIDCOM requirements [RFC3304] defines the general security requirements for the MIDCOM protocol. The SNMPv3 User-based Security Model (USM, [RFC2574]) satisfies those requirements. USM defines three standardized methods for providing authentication, confidentiality, and integrity. The method to use can be optionally chosen. The methods operate securely across untrusted domains. Additionally, USM has specific built-in mechanisms for preventing replay attacks including unique protocol engine IDs, timers and counters per engine and time windows for the validity of messages.

7. Changes since last version

The following summarizes the major changes made from the 00 in creating the 01 version of the WG document:

- Added Appendix A, providing a working version of a detailed analysis of the NATMIB vs. the MIDCOM semantics. Updated section 4.1 to reflect the analysis of the NATMIB to date.
- Summarized why there are 2 MIDCOM MIB proposals and the issues related to the development of the MIDCOM MIB from design team discussions and the WG mailing list MIDCOM semantics discussions (section 5).
- Updated the current status of the IPSEC Policy Configuration MIB as it applies to MIDCOM (section 4.3).

The following summarizes the major changes made from the 01 individual draft for the 00 WG document:

- Changed the focus of the document to be the identification of the managed objects rather than the actual MIDCOM MIB. Any new MIB modules required will be detailed in a separate document upon completion of this analysis. This change does not impact the majority of the content as the only content thus far has been the analysis aspects.

The following summarizes the major changes made to the 01 version of the individual document from the previous version (draft-barnes-midcom-mib-00):

- Miscellaneous editorial changes include basic formatting and changing references of mib to MIB, and mibs to MIB modules.
- Removed reference to SNMP proxy functionality as that's not applicable to MIDCOM.
- Updated references to include additional informational references for Diffserv and updated versions on some drafts.
- Incorporated "Protocol" into the title of the document.
- In general, attempted to clarify references to policy to be specific to the rulesets as they apply to a session.
- Some minor re-arranging of text in section 2 to try to improve the readability of the document.
- Clarified that the configuration relevant to MIDCOM is primarily dynamic.
- Removed some of the non-relevant text in sections 3 (eg. References to CLI in the configuration section and some details in the Policy Coordination Section). Totally removed the Policy Specification section since it is out of scope.
- Section 4: Added analysis on Diffserv MIB, IPSEC Policy Config MIB and Policy Based Management MIB.

Normative References

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

[RFC3303] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, A. Rayhan, "Middlebox Communications Architecture and Framework", RFC 3303, August, 2002.

[MDCSEM] Stiemerling, M., Quittek, J., Taylor, T., "MIDCOM Protocol Semantics", draft-ietf-midcom-semantics-05.txt, October, 2003.

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

[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.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", STD 62, RFC 3411, November 2002.

[RFC3412] Case, J., Harrington D., Presuhn R., and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3412, November 2002.

[RFC3413] Levi, D., Meyer, P., and B. Stewart, "SNMPv3 Applications", STD 62, RFC 3413, November 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, November 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, November 2002.

[NATMIB] Raghunarayan, R., Pai, N., Rohit, R., Wang, C., Srisuresh, P., "Definitions of Managed Objects for Network Address Translators (NAT)", draft-ietf-nat-natmib-06.txt, September, 2003.

[PBMMIB] Waldbusser, S., Saperia, J., Hongal, T., "Policy Based Management MIB", draft-ietf-snmpconf-pm-13.txt, March, 2003.

[IPCMIB] Baer, M., Charlet, R., Hardaker, W., Story, R., Wang, C., "IPsec Policy Configuration MIB module", draft-ietf-ipsp-ipsec-conf-MIB-06.txt, March, 2003.

[MDCMIB] TBD.

Informative References

[RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", 3410, November 2002.

[MDCPEV] Barnes, M., "Middlebox Communications (MIDCOM) Protocol Evaluation", draft-ietf-midcom-protocol-eval-06.txt, November, 2002.

[RFC2287] Krupczak, C. and J. Saperia, "Definitions of System-Level Managed Objects for Applications", RFC 2287, February 1998.

[RFC 2475] Blake, S., et al, "An Architecture for Differentiated Service", RFC 2475, December 1998.

[RFC2564] C. Kalbfleisch, C. Krupczak, R.Presuhn, J. Saperia, "Application Management MIB", May 1999.

[RFC2594] H. Hazewinkel, C. Kalbfleisch, J. Schoenwaelder, "Definitions of Managed Objects for WWW Services", May 1999.

[RFC2663] P. Srisuresh, M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations", August 1999.

[RFC2788] N. Freed, S. Kille, "Network Services Monitoring MIB", RFC 2788, March 2000.

[RFC2790] S. Waldbusser, P. Grillo, "Host Resources MIB", March 2000.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB using SMIv2", RFC 2863, June 2000.

[RFC3289] Baker, F., Chan, K., Smith, A., "Management Information Base for the Differentiated Services Architecture", RFC 3289, May 2002.

[RFC3290] Bernet, Y., et al, "An Informal Management Model for Differentiated Services Routers", RFC 3290, May 2002.

[DPCMIB] Hazewinkel, H, Partain, D., "The Differentiated Services Configuration MIB", draft-ietf-snmpconf-diffpolicy-05.txt, June 2002.

[BRGMIB] Norseth, K.C. and Bell, E., "Definitions of Managed Objects for Bridges", draft-ietf-bridge-bridgeMIB-smiv2-04.txt, October 2002.

[BREMIB] Ngai, V., "Definitions of Managed Objects for Bridges with Traffic Classes, Multicast Filtering and Virtual LAN Extensions", draft-ietf-bridge-ext-v2-01.txt, September 2002.

[81xMIB] Norseth, K.C. "Definitions for Port Access Control (IEEE 802.1X) MIB", draft-ietf-bridge-8021x-01.txt, February, 2003.

Acknowledgements

The authors would like to thank Randy Presuhn and Pyda Srisuresh for their comments and feedback on the initial version of this document.

Contributors' Addresses

The following individuals participated in the MIDCOM MIB design team and thus provided implicit and explicit content for this document:

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

EMail: hardaker@tislabs.com

David Harrington, Co-chair SNMPv3 WG
Enterasys Networks
35 Industrial Way
Rochester, NH 03867-5005
USA

Phone: +1 603-337-2614
EMail: dbh@enterasys.com

Juergen Quittek

NEC Europe Ltd.
Network Laboratories
Kurfuersten-Anlage 36
69115 Heidelberg
Germany

Phone: +49 6221 90511-15
EMail: quittek@ccrle.nec.de

Martin Stiemerling
NEC Europe Ltd.
Network Laboratories
Adenauerplatz 6
69115 Heidelberg
Germany

Phone: +49 6221 90511-13
Email: stiemerling@ccrle.nec.de

Tom Taylor
Nortel Networks
1852 Lorraine Ave.
Ottawa, Ontario
Canada K1H 6Z8

Phone: +1 613 736 0961
Email: taylor@nortelnetworks.com

Editor's Address

Mary Barnes
Nortel Networks
2380 Performance Drive
Richardson, TX 75082
USA

Phone: 1-972-684-5432
Email: mbarnes@nortelnetworks.com

Appendix A Analysis of the NATMIB against the MIDCOM semantics

This is an analysis of how the MIDCOM semantics requests and notifications could be reflected by changes in the NAT MIB, and vice versa. It compares [MDCSEM] with [NATMIB]. It does not reflect a consensus of the design team, but rather is put forth as a position (authored by Tom Taylor, edited by Mary) for discussion and presents

some issues requiring resolution in order to fully understand the impact on the applicability of the NATMIB to MIDCOM from the perspective of the MIDCOM semantics.

A.1 NAT MIB Summary

The NAT MIB as defined in draft-ietf-nat-natmib-06.txt consists of the following tables. Based upon design team discussions, the group and owner will be deleted from the next version of the NAT MIB, wherever they appear. The following annotation applies to the summary:
RC = Read-Create; RO = Read-Only

Config: Interface table
Index
Public/private RC
Type of NAT across this interface (basic, NAPT, bidirectional, twice-NAT) RC
Map name (for all maps relating to this interface) RC
Housekeeping stuff (storage type, active/inactive) RC

Config: Address map table
Map name (as given in interface table)
Map sub-index
OwnerID RC
GroupID RO
Static/dynamic entry type RC
Trigger for mapping: inbound source, inbound destination, outbound source, outbound destination RC
Local address/port range RC
Global address/port range RC
UDP/TCP/ICMP/other RC
Housekeeping stuff (storage type, active/inactive) RC

Translation: Address bind table
Local (private) address type and value (index)
Owner RO
Group RO
Global (public) address type and value RC
BindID RO
Unidirectional/bidirectional
same as underlying address map entry RC
Static/dynamic RC
Map name in address map table RC
Number of active sessions using this bind RO
Maximum life of bind while no sessions active RC
Accumulated idle time with no sessions active RO

```
  Number of inbound packets successfully translated using this bind
  entry RO
  Number of inbound packets successfully translated using this bind
  entry RO
  Active/inactive status RC

Translation: Address-port bind table (applies to NAPT)
  Local (private) address type and value (index)
  Local port.  For ICMP query-id is used instead of port. (index)
  Protocol.  "None" => all IP. (index)
  Owner RO
  Group RO
  Global (public) address type and value RC
  Global port RC
  BindID -- same as in address bind table RO
  Unidirectional/bidirectional -- same as underlying address map
  entry RC
  Static/dynamic RC
  Map name in address map table RC
  Number of active sessions using this bind RO
  Maximum life of bind while no sessions active RC
  Accumulated idle time with no sessions active RO
  Number of inbound packets successfully translated using this bind
  entry RO
  Number of inbound packets successfully translated using this bind
  entry RO
  Active/inactive status RC

Translation: session table
  BindID in bind tables (index)
  SessionId (index)
  Owner RO
  Group RO
  In/out (relative to private network) RC
  Session up time RO
  Protocol RC
  Original private address type and port (A0) RC
  Translated private address type and port (A2) RC
  Original public address type and port (A3) RC
  Translated public address type and port (A1) RC
  Maximum life of session while no packets detected RC
  Accumulated idle time since last packet detected RO
  Other bindID in case of twice-NAT RC
  Number of inbound packets successfully translated in this session
  RO
  Number of inbound packets successfully translated in this session
  RO
  Active/inactive status RC
```

A.2 General remarks

E-mail discussion has indicated that an interface is a logical point of attachment to a given subnetwork.

There is debate over whether it is possible to have flows from one private interface to another. The [NATMIB] authors did not expect that this could happen and it is not described in [RFC2663], for example. The MIB structure always assumes that flows are between a private and a public interface.

The NATMIB seems to provide redundant mapping capability, in that the same mapping can be specified either on the public or on the private interface through which a given flow passes. The assumption is, based on the comments for the NATMIB natConfAddrMapTranslationEntity, that the mapping is always specified on the public side for the translation between the interior endpoint address A0 and the intermediate exterior address A2. For twice-NAT, the mapping between the exterior endpoint address A3 and the interior endpoint address A1 is specified on the private interface. (A0, A1, A2, and A3 as described in [MDCSEM]. These may designate a range of ports as well as an address, depending on the protocol.)

A further note from the same comments in the NATMIB is that "inbound" is always toward the NAT, "outbound" is always away from it. The specific example given in the comments is a session flow from a private to a public interface. On the public interface, it would be seen in the NATMIB as "outbound", while on the private interface it is seen as "inbound".

The difference between the address map and the bind tables appears to be as follows:
- the bind tables deal with mapping between specific address/port pairs rather than ranges;
- a bind may be automatically deleted after a certain amount of time without an active associated session;
- certain statistics are captured.

It is not clear how the unidirectional/bidirectional setting for a bind entry is derived from the parent address map. One possibility is that it comes from the type of NAT service provided on the interface. The other is that it is determined by seeing if the address map includes sub-entries for both inbound and outbound triggers.

A.3 Interaction between PRR and the NAT MIB

[RFC2663] (NAT terminology) does not distinguish between mapping and binding, saying only that binding is the first step in the NAT

translation process. This may have led to some confusion in discussions amongst design team members, depending on what was meant by "binding". The key question is whether an address map is allowed to exist with no bindings referring to it. If this is possible only in some implementations or is generally impossible, then PRR MUST affect both the address map table and the address and address-port bind tables. Otherwise, it can be argued that PRR should just affect the address map table. As indicated below, this allows the MIDCOM MIB to implement the MIDCOM concept of a fixed rule lifetime, in contrast to (and over-riding) the traditional NAT concept of maximum idle time.

PRR requests the reservation of an external address A2 to which the internal endpoint address A0 will be mapped, and in the twice-NAT case also asks for the reservation of an internal address A1 on the interface to which A0 is connected. The unspecified external address A3 will be mapped to A1. The relevant parameters of PRR are:

- service type: traditional or twice-NAT
- transport protocol
- internal address type and value (potentially wildcarded, if allowed by the NAT implementation)
- internal starting port number and range
- parity of the starting external port
- external address type
- internal (private) interface (optional)
- external (public) interface (optional)
- requested reservation lifetime.

groupID has been omitted from this list because it is a MIDCOM rather than a NAT issue. The requested reservation lifetime is not an existing NAT concept, but has an important bearing on how the PRR is reflected in the NAT MIB, per the arguments below.

The response to PRR includes the assigned external address(es) and port(s).

A.3.1 Preliminary Comments

According to the NAT MIB, the NAT type is a configured property of the interface and therefore cannot be arbitrarily chosen. It is for this reason that this analysis suggests that the service type parameter should not be present in the PRR.

The internal interface parameter should only be needed if the MIDCOM agent is acting on behalf of an internal endpoint, which is served by a different interface. Otherwise it would be expected that the middlebox would use the internal interface on which the PRR arrived.

It's not clear how the MIDCOM agent would know the applicable interface identifier and whether it has to use it. It seems there is a good argument for expecting that the MIDCOM agent and the internal endpoint are served by the same internal interface, simply because the MIDCOM agent is, by assumption, on the signalling path for the application.

The value of having the external interface parameter isn't apparent. Either the internal address-port A0 is already covered by a binding or it is not. If it is, the external address is determined by the existing bind. If it is not, it should be up to the middlebox to choose the external interface, based on existing mapping table entries, load balancing considerations, etc.

The possibility that A0 is a range of addresses rather than a single address is problematic. The problem comes about if different addresses within the range are covered by different existing bindings or map entries pointing to different external addresses. It's anticipated that it will be necessary to have a return code indicating that assignment could not be made due to conflict. Alternatively we can disallow wildcarding of A0 -- something that would seem to make sense in terms of application requirements.

We have to accept the possibility that a PER following a PRR may fail. This may be because the middlebox has selected an external interface at the PRR stage through which A3 turns out to be unreachable. The second possibility is in the twice-NAT case and has been discussed already on the list: A3 when it becomes known may turn out to be contained in a binding that conflicts with the mapping to internal address A1.

A.3.2 Choosing A1 and A2

The PRR must always result in the assignment of an intermediate external address A2. It attempts to select A2 (address and ports, if applicable) according to the following possibilities in decreasing order of preference:

(1) protocol and local address A0 from the PRR match an active bind on any public interface. In this case, the address part of A2 is determined uniquely by the bind; the port part may be determined by active bind(s), may conflict with active binds, or may have to be chosen arbitrarily by the middlebox to match the requested range and parity. Conflict occurs if existing binds prevent the assignment of consecutive ports in the requested

range or disagree with the requested parity. The PRR fails as a result.

(2) local address A0 appears in an address map entry on any public interface and, if applicable, the internal port range in the PRR is included in the local port range for the map entry. In this case, the address part of A2 may be determined uniquely by the address map entry or may have to be selected from the possible range, and similarly for the ports. There is the remote possibility of a parity conflict if port mapping is determined uniquely but is of the wrong parity.

(3) no applicable bind or address map entry is found. In this case the middlebox assigns A2, address and ports, based on internal logic.

If the NAT type on the internal interface is "twice-NAT", the PRR must also result in the assignment of an intermediate internal address A1 on that interface. The middlebox selects address A1 (and ports, if applicable) from eligible possibilities for the internal interface based on internal logic.

A.3.3 Effect of Assigning A1 and A2 On the NAT MIB

Once A2 (and A1, if applicable) have been determined, the middlebox must create address map entries to support them. This is where the concept of reservation life is significant. Since the operation of the requested reservation lifetime is different from the maximum idle time assigned to binds, it seems best implemented by making the address map entries static. The lifetime is then controlled and reflected by the MIDCOM MIB rather than the NAT MIB.

Creating new address map entries is straightforward if none previously existed. The question is what to do when an applicable address map entry already exists. The alternatives are to do nothing, to create a new entry overlapping with the existing entry, or else split the existing entry into multiple entries, including one matching the assigned address and ports and as many others as necessary to cover the original range.

Relying on the existing entry alone ("do nothing") requires that a reference count be attached to the address map entry in the NAT MIB to record the number of MIDCOM policy rules relying on it. This count is increased by 1 if the original entry was created by means other than a MIDCOM request. The count is reduced by 1 each time a MIDCOM policy rule lifetime expires or the rule is deleted, or if the non-MIDCOM process originally responsible for creating the address

map entry decides to delete it. When the count reaches zero the address map entry, all child binds, and all sessions deriving from those binds are deleted. One problem with the "do nothing" approach is that if the original process creating the entry was not a MIDCOM request and no longer requires the entry, the scope of the entry (address range, port range) will in general be broader than required by the dependent MIDCOM policy rules. As will be seen when the MIDCOM query primitives are discussed, this approach will require some redundancy between the MIDCOM and NAT MIBs.

The overlapping entry approach avoids the need for reference counts and provides a precise match to each MIDCOM request. However, the non-MIDCOM entries must be flagged to ensure that they are the ones non-MIDCOM processes use to create binds. The MIDCOM MIB records the map name and sub-index of the address map entry corresponding to a given policy rule, so no policy rule linking information is be needed in the NAT MIB itself. This approach has the advantage that it avoids the redundancy referred to in the previous paragraph.

The entry split approach is not really practical. To ensure that deletions by MIDCOM and by non-MIDCOM processes were handled correctly, the fragments of the original range would have to be identified by the original address map entry sub-index as well as a new one for each fragment. Reference counts might also be needed. Moreover, the number of address map entries would grow more rapidly than with the overlapping address map entry approach.

<Analysis of other MIDCOM primitives to be added.>

Full Copyright Statement

assigns. This document and the information contained