

Network Working Group
Request For Comments: 2742
Category: Standards Track

L. Heintz
Cisco Systems
S. Gudur
Independent Consultant
M. Ellison, Ed.
Ellison Software Consulting, Inc.
January 2000

Definitions of Managed Objects for Extensible SNMP Agents

Status of this Memo

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

Copyright Notice

Copyright (C) The Internet Society (2000). All Rights Reserved.

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 objects managing SNMP agents that use the Agent Extensibility (AgentX) Protocol.

This memo specifies a MIB module in a manner that is both compliant to the SMIV2, and semantically identical to the peer SMIV1 definitions.

Table of Contents

1. The SNMP Management Framework	2
2. Introduction	3
3. AgentX MIB Overview	3
4. Managed Object Definitions for AgentX	4
5. Intellectual Property	15
6. Acknowledgements	16
7. Security Considerations	16
8. References	17
9. Authors' and Editor's Addresses	19
10. Full Copyright Statement	20

1. The SNMP Management Framework

The SNMP Management Framework presently consists of five major components:

- An overall architecture, described in RFC 2571 [1].
- Mechanisms for describing and naming objects and events for the purpose of management. The first version of this Structure of Management Information (SMI) is called SMIV1 and described in STD 16, RFC 1155 [2], STD 16, RFC 1212 [3] and RFC 1215 [4]. The second version, called SMIV2, is described in STD 58, RFC 2578 [5], STD 58, RFC 2579 [6] and STD 58, RFC 2580 [7].
- Message protocols for transferring management information. The first version of the SNMP message protocol is called SNMPv1 and described in STD 15, RFC 1157 [8]. A second version of the SNMP message protocol, which is not an Internet standards track protocol, is called SNMPv2c and described in RFC 1901 [9] and RFC 1906 [10]. The third version of the message protocol is called SNMPv3 and described in RFC 1906 [10], RFC 2572 [11] and RFC 2574 [12].
- Protocol operations for accessing management information. The first set of protocol operations and associated PDU formats is described in STD 15, RFC 1157 [8]. A second set of protocol operations and associated PDU formats is described in RFC 1905 [13].
- A set of fundamental applications described in RFC 2573 [14] and the view-based access control mechanism described in RFC 2575 [15].

A more detailed introduction to the current SNMP Management Framework can be found in RFC 2570 [16].

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. Objects in the MIB are defined using the mechanisms defined in the SMI.

This memo specifies a MIB module that is compliant to the SMIV2. A MIB conforming to the SMIV1 can be produced through the appropriate translations. The resulting translated MIB must be semantically equivalent, except where objects or events are omitted because no translation is possible (use of Counter64). Some machine readable information in SMIV2 will be converted into textual descriptions in

SMIv1 during the translation process. However, this loss of machine readable information is not considered to change the semantics of the MIB.

2. Introduction

The SNMP Agent Extensibility Protocol (AgentX) is a protocol used to distribute the implementation of an SNMP agent amongst a single "master agent" and multiple "subagents". See [17] for details about the AgentX protocol.

The goals of the AgentX MIB are:

- List the set of subagent connections that currently have logical sessions open with the master agent.
- Identify each subagent connection transport address and type.
- Identify each subagent session vendor, AgentX protocol version, and other characteristics.
- Identify the set of MIB objects each session implements, the context in which the objects are registered, and the priority of the registration.
- Determine protocol operational parameters such as the timeout interval for responses from a session and the priority at which a session registers a particular MIB region.
- Allow (but do not require) managers to explicitly close subagent sessions with the master agent.

3. AgentX MIB Overview

This MIB is organized into four groups. The agentxGeneral group provides information describing the master agent's AgentX support, including the protocol version supported. The agentxConnection group provides information describing the current set of connections capable of carrying AgentX sessions. The agentxSession group provides information describing the current set of AgentX sessions. The agentxRegistration group provides information describing the current set of registrations.

Three tables form the heart of this mib. These are the connection, session, and registration tables.

Entries in the registration table exist in a many-to-one relationship with entries in the session table. This relationship is expressed through the two common indices, agentxSessionIndex and agentxConnIndex. Entries in the registration table also exist in a many-to-one relationship with entries in the connection table. This relationship is expressed through the common index, agentxConnIndex.

Entries in the session table exist in a many-to-one relationship with entries in the connection table. This relationship is expressed through the common index, agentxConnIndex.

4. Managed Object Definitions for AgentX

```
AGENTX-MIB DEFINITIONS ::= BEGIN
```

```
IMPORTS
```

```
  MODULE-IDENTITY, OBJECT-TYPE, Unsigned32, mib-2
    FROM SNMPv2-SMI
```

```
  SnmpAdminString
    FROM SNMP-FRAMEWORK-MIB
```

```
  MODULE-COMPLIANCE, OBJECT-GROUP
    FROM SNMPv2-CONF
```

```
  TEXTUAL-CONVENTION, TimeStamp, TruthValue, TDomain
    FROM SNMPv2-TC;
```

```
agentXMIB MODULE-IDENTITY
```

```
  LAST-UPDATED "200001100000Z" -- Midnight 10 January 2000
```

```
  ORGANIZATION "AgentX Working Group"
```

```
  CONTACT-INFO "WG-email:   agentx@dorothy.bmc.com
                Subscribe:  agentx-request@dorothy.bmc.com
                WG-email Archive: ftp://ftp.peer.com/pub/agentx/archives
                FTP repository: ftp://ftp.peer.com/pub/agentx
                http://www.ietf.org/html.charters/agentx-charter.html"
```

```
  Chair:      Bob Natale
                ACE*COMM Corporation
  Email:      bnatale@acecomm.com
```

```
  WG editor:  Mark Ellison
                Ellison Software Consulting, Inc.
  Email:      ellison@world.std.com
```

```
  Co-author:  Lauren Heintz
                Cisco Systems,
  EMail:      lheintz@cisco.com
```

```
  Co-author:  Smitha Gudur
                Independent Consultant
  Email:      sgudur@hotmail.com
```

```
"
DESCRIPTION      "This is the MIB module for the SNMP Agent Extensibility
                  Protocol (AgentX).  This MIB module will be implemented by
                  the master agent.
"

REVISION         "200001100000Z" -- Midnight 10 January 2000
DESCRIPTION      "Initial version published as RFC 2742."

 ::= { mib-2 74 }

-- Textual Conventions

AgentxTAddress ::= TEXTUAL-CONVENTION
STATUS          current
DESCRIPTION     "Denotes a transport service address.  This is identical to
                  the TAddress textual convention (SNMPv2-SMI) except that
                  zero-length values are permitted.
"
SYNTAX          OCTET STRING (SIZE (0..255))

-- Administrative assignments

agentxObjects OBJECT IDENTIFIER ::= { agentxMIB 1 }
agentxGeneral OBJECT IDENTIFIER ::= { agentxObjects 1 }
agentxConnection OBJECT IDENTIFIER ::= { agentxObjects 2 }
agentxSession OBJECT IDENTIFIER ::= { agentxObjects 3 }
agentxRegistration OBJECT IDENTIFIER ::= { agentxObjects 4 }

agentxDefaultTimeout OBJECT-TYPE
SYNTAX          INTEGER (0..255)
UNITS           "seconds"
MAX-ACCESS     read-only
STATUS          current
DESCRIPTION     "The default length of time, in seconds, that the master
                  agent should allow to elapse after dispatching a message
                  to a session before it regards the subagent as not
                  responding.  This is a system-wide value that may
                  override the timeout value associated with a particular
                  session (agentxSessionTimeout) or a particular registered
                  MIB region (agentxRegTimeout).  If the associated value of
                  agentxSessionTimeout and agentxRegTimeout are zero, or
                  impractical in accordance with implementation-specific
                  procedure of the master agent, the value represented by
                  this object will be the effective timeout value for the
```

```

        master agent to await a response to a dispatch from a
        given subagent.
    "
    DEFVAL          { 5 }
    ::= { agentxGeneral 1 }

agentxMasterAgentXVer OBJECT-TYPE
    SYNTAX          INTEGER (1..255)
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The AgentX protocol version supported by this master agent.
        The current protocol version is 1. Note that the master agent
        must also allow interaction with earlier version subagents.
    "
    ::= { agentxGeneral 2 }

--          The AgentX Subagent Connection Group

agentxConnTableLastChange OBJECT-TYPE
    SYNTAX          TimeStamp
    MAX-ACCESS      read-only
    STATUS          current
    DESCRIPTION
        "The value of sysUpTime when the last row creation or deletion
        occurred in the agentxConnectionTable.
    "
    ::= { agentxConnection 1 }

agentxConnectionTable OBJECT-TYPE
    SYNTAX          SEQUENCE OF AgentxConnectionEntry
    MAX-ACCESS      not-accessible
    STATUS          current
    DESCRIPTION
        "The agentxConnectionTable tracks all current AgentX transport
        connections. There may be zero, one, or more AgentX sessions
        carried on a given AgentX connection.
    "
    ::= { agentxConnection 2 }

agentxConnectionEntry OBJECT-TYPE
    SYNTAX          AgentxConnectionEntry
    MAX-ACCESS      not-accessible
    STATUS          current

    DESCRIPTION
        "An agentxConnectionEntry contains information describing a
        single AgentX transport connection. A connection may be

```

used to support zero or more AgentX sessions. An entry is created when a new transport connection is established, and is destroyed when the transport connection is terminated.

"

```
INDEX { agentxConnIndex }
 ::= { agentxConnectionTable 1 }
```

```
AgentxConnectionEntry ::= SEQUENCE {
    agentxConnIndex      Unsigned32,
    agentxConnOpenTime   TimeStamp,
    agentxConnTransportDomain TDomain,
    agentxConnTransportAddress AgentxTAddress }
```

agentxConnIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"agentxConnIndex contains the value that uniquely identifies an open transport connection used by this master agent to provide AgentX service. Values of this index should not be re-used. The value assigned to a given transport connection is constant for the lifetime of that connection.

"

```
 ::= { agentxConnectionEntry 1 }
```

agentxConnOpenTime OBJECT-TYPE

SYNTAX TimeStamp

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The value of sysUpTime when this connection was established and, therefore, its value when this entry was added to the table.

"

```
 ::= { agentxConnectionEntry 2 }
```

agentxConnTransportDomain OBJECT-TYPE

SYNTAX TDomain

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The transport protocol in use for this connection to the subagent.

"

```
 ::= { agentxConnectionEntry 3 }
```

agentxConnTransportAddress OBJECT-TYPE

SYNTAX AgentxTAddress

```

MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The transport address of the remote (subagent) end of this
    connection to the master agent.  This object may be zero-length
    for unix-domain sockets (and possibly other types of transport
    addresses) since the subagent need not bind a filename to its
    local socket.
    "
 ::= { agentxConnectionEntry 4 }

-- The AgentX Subagent Session Group

agentxSessionTableLastChange OBJECT-TYPE
SYNTAX        TimeStamp
MAX-ACCESS    read-only
STATUS        current
DESCRIPTION
    "The value of sysUpTime when the last row creation or deletion
    occurred in the agentxSessionTable.
    "
 ::= { agentxSession 1 }

agentxSessionTable OBJECT-TYPE
SYNTAX        SEQUENCE OF AgentxSessionEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "A table of AgentX subagent sessions currently in effect.
    "
 ::= { agentxSession 2 }

agentxSessionEntry OBJECT-TYPE
SYNTAX        AgentxSessionEntry
MAX-ACCESS    not-accessible
STATUS        current
DESCRIPTION
    "Information about a single open session between the AgentX
    master agent and a subagent is contained in this entry.  An
    entry is created when a new session is successfully established
    and is destroyed either when the subagent transport connection
    has terminated or when the subagent session is closed.
    "
INDEX          { agentxConnIndex, agentxSessionIndex }
 ::= { agentxSessionTable 1 }

AgentxSessionEntry ::= SEQUENCE {
    agentxSessionIndex      Unsigned32,

```

```

agentxSessionObjectID OBJECT IDENTIFIER,
agentxSessionDescr SnmpAdminString,
agentxSessionAdminStatus INTEGER,
agentxSessionOpenTime TimeStamp,
agentxSessionAgentXVer INTEGER,
agentxSessionTimeout INTEGER
}

```

agentxSessionIndex OBJECT-TYPE

SYNTAX Unsigned32 (0..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"A unique index for the subagent session. It is the same as h.sessionID defined in the agentx header. Note that if a subagent's session with the master agent is closed for any reason its index should not be re-used. A value of zero(0) is specifically allowed in order to be compatible with the definition of h.sessionId.

"

::= { agentxSessionEntry 1 }

agentxSessionObjectID OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"This is taken from the o.id field of the agentx-Open-PDU. This attribute will report a value of '0.0' for subagents not supporting the notion of an AgentX session object identifier.

"

::= { agentxSessionEntry 2 }

agentxSessionDescr OBJECT-TYPE

SYNTAX SnmpAdminString

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"A textual description of the session. This is analogous to sysDescr defined in the SNMPv2-MIB in RFC 1907 [19] and is taken from the o.descr field of the agentx-Open-PDU. This attribute will report a zero-length string value for subagents not supporting the notion of a session description.

"

::= { agentxSessionEntry 3 }

agentxSessionAdminStatus OBJECT-TYPE

```

SYNTAX      INTEGER {
                up(1),
                down(2)
            }
MAX-ACCESS  read-write
STATUS      current
DESCRIPTION
    "The administrative (desired) status of the session.  Setting
    the value to 'down(2)' closes the subagent session (with c.reason
    set to 'reasonByManager')."
"
 ::= { agentxSessionEntry 4 }

```

agentxSessionOpenTime OBJECT-TYPE

```

SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of sysUpTime when this session was opened and,
    therefore, its value when this entry was added to the table."
"
 ::= { agentxSessionEntry 5 }

```

agentxSessionAgentXVer OBJECT-TYPE

```

SYNTAX      INTEGER (1..255)
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The version of the AgentX protocol supported by the
    session.  This must be less than or equal to the value of
    agentxMasterAgentXVer."
"
 ::= { agentxSessionEntry 6 }

```

agentxSessionTimeout OBJECT-TYPE

```

SYNTAX      INTEGER (0..255)
UNITS       "seconds"
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The length of time, in seconds, that a master agent should
    allow to elapse after dispatching a message to this session
    before it regards the subagent as not responding.  This value
    is taken from the o.timeout field of the agentx-Open-PDU.
    This is a session-specific value that may be overridden by
    values associated with the specific registered MIB regions
    (see agentxRegTimeout).  A value of zero(0) indicates that
    the master agent's default timeout value should be used

```

```

        (see agentxDefaultTimeout).
    "
 ::= { agentxSessionEntry 7 }

-- The AgentX Registration Group

agentxRegistrationTableLastChange OBJECT-TYPE
SYNTAX      TimeStamp
MAX-ACCESS  read-only
STATUS      current
DESCRIPTION
    "The value of sysUpTime when the last row creation or deletion
    occurred in the agentxRegistrationTable.
    "
 ::= { agentxRegistration 1 }

agentxRegistrationTable OBJECT-TYPE
SYNTAX      SEQUENCE OF AgentxRegistrationEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "A table of registered regions.
    "
 ::= { agentxRegistration 2 }

agentxRegistrationEntry OBJECT-TYPE
SYNTAX      AgentxRegistrationEntry
MAX-ACCESS  not-accessible
STATUS      current
DESCRIPTION
    "Contains information for a single registered region. An
    entry is created when a session successfully registers a
    region and is destroyed for any of three reasons: this region
    is unregistered by the session, the session is closed,
    or the subagent connection is closed.
    "
INDEX       { agentxConnIndex, agentxSessionIndex, agentxRegIndex }
 ::= { agentxRegistrationTable 1 }

AgentxRegistrationEntry ::= SEQUENCE {
agentxRegIndex      Unsigned32,
agentxRegContext    OCTET STRING,
agentxRegStart      OBJECT IDENTIFIER,
agentxRegRangeSubId Unsigned32,
agentxRegUpperBound Unsigned32,
agentxRegPriority    Unsigned32,
agentxRegTimeout    INTEGER,
agentxRegInstance   TruthValue }

```

agentxRegIndex OBJECT-TYPE

SYNTAX Unsigned32 (1..4294967295)

MAX-ACCESS not-accessible

STATUS current

DESCRIPTION

"agentxRegIndex uniquely identifies a registration entry.

This value is constant for the lifetime of an entry.

"

::= { agentxRegistrationEntry 1 }

agentxRegContext OBJECT-TYPE

SYNTAX OCTET STRING

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The context in which the session supports the objects in this region. A zero-length context indicates the default context.

"

::= { agentxRegistrationEntry 2 }

agentxRegStart OBJECT-TYPE

SYNTAX OBJECT IDENTIFIER

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"The starting OBJECT IDENTIFIER of this registration entry. The session identified by agentxSessionIndex implements objects starting at this value (inclusive). Note that this value could identify an object type, an object instance, or a partial object instance.

"

::= { agentxRegistrationEntry 3 }

agentxRegRangeSubId OBJECT-TYPE

SYNTAX Unsigned32

MAX-ACCESS read-only

STATUS current

DESCRIPTION

"agentxRegRangeSubId is used to specify the range. This is taken from r.region_subid in the registration PDU. If the value of this object is zero, no range is specified. If it is non-zero, it identifies the 'nth' sub-identifier in r.region for which this entry's agentxRegUpperBound value is substituted in the OID for purposes of defining the region's upper bound.

"

::= { agentxRegistrationEntry 4 }

agentxRegUpperBound OBJECT-TYPE

SYNTAX Unsigned32
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"agentxRegUpperBound represents the upper-bound sub-identifier in a registration. This is taken from the r.upper_bound in the registration PDU. If agentxRegRangeSubid (r.region_subid) is zero, this value is also zero and is not used to define an upper bound for this registration.

"

```
::= { agentxRegistrationEntry 5 }
```

agentxRegPriority OBJECT-TYPE

SYNTAX Unsigned32
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"The registration priority. Lower values have higher priority. This value is taken from r.priority in the register PDU. Sessions should use the value of 127 for r.priority if a default value is desired.

"

```
::= { agentxRegistrationEntry 6 }
```

agentxRegTimeout OBJECT-TYPE

SYNTAX INTEGER (0..255)
 UNITS "seconds"
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"The timeout value, in seconds, for responses to requests associated with this registered MIB region. A value of zero(0) indicates the default value (indicated by by agentxSessionTimeout or agentxDefaultTimeout) is to be used. This value is taken from the r.timeout field of the agentx-Register-PDU.

"

```
::= { agentxRegistrationEntry 7 }
```

agentxRegInstance OBJECT-TYPE

SYNTAX TruthValue
 MAX-ACCESS read-only
 STATUS current

DESCRIPTION

"The value of agentxRegInstance is 'true' for registrations for which the INSTANCE_REGISTRATION was set, and is 'false' for all other registrations.

"

```

 ::= { agentxRegistrationEntry 8 }

-- Conformance Statements for AgentX

agentxConformance      OBJECT IDENTIFIER ::= { agentxMIB 2 }
agentxMIBGroups        OBJECT IDENTIFIER ::= { agentxConformance 1 }
agentxMIBCompliances   OBJECT IDENTIFIER ::= { agentxConformance 2 }

-- Compliance Statements for AgentX

agentxMIBCompliance MODULE-COMPLIANCE
STATUS      current
DESCRIPTION
    "The compliance statement for SNMP entities that implement the
    AgentX protocol. Note that a compliant agent can implement all
    objects in this MIB module as read-only.
    "
MODULE -- this module
    MANDATORY-GROUPS { agentxMIBGroup }

    OBJECT agentxSessionAdminStatus
        MIN-ACCESS read-only
        DESCRIPTION
            "Write access is not required.
            "
 ::= { agentxMIBCompliances 1 }

agentxMIBGroup OBJECT-GROUP
OBJECTS {
    agentxDefaultTimeout,
    agentxMasterAgentXVer,
    agentxConnTableLastChange,
    agentxConnOpenTime,
    agentxConnTransportDomain,
    agentxConnTransportAddress,
    agentxSessionTableLastChange,
    agentxSessionTimeout,
    agentxSessionObjectID,
    agentxSessionDescr,
    agentxSessionAdminStatus,
    agentxSessionOpenTime,
    agentxSessionAgentXVer,
    agentxRegistrationTableLastChange,
    agentxRegContext,
    agentxRegStart,
    agentxRegRangeSubId,
    agentxRegUpperBound,
    agentxRegPriority,

```

```
    agentxRegTimeout,
    agentxRegInstance
  }
STATUS      current
DESCRIPTION
  "All accessible objects in the AgentX MIB.
  "
 ::= { agentxMIBGroups 1 }

END
```

5. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in BCP-11. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

6. Acknowledgements

This document is the result of the efforts of the IETF AgentX Working Group (WG).

This MIB is an evolution of the Subagent MIB by Bert Wijnen (wijnen@vnet.ibm.com) which in turn was derived from the SMUX-MIB by Marshall Rose [18].

Thanks are in order to the following AgentX WG members:

- Mike Daniele (Compaq Computer Corporation)
- Dale Francisco (Cisco Systems)
- Bob Natale (ACE*COMM Corporation)
- Randy Presuhn (BMC Software, Inc.)
- Shawn Routhier (Epilogue)
- Mike Thatcher (Independent Consultant)

Special acknowledgement is made to:

- Maria Greene (Xedia)

Special acknowledgement is also made to the following individuals for participating in the 1998 AgentX testing summit (bakeoff) held in Sunnyvale, California:

- Jeff Case (SNMP Research, Inc.)
- Mike Daniele (Compaq Computer Corporation)
- Mark Ellison (Ellison Software Consulting, Inc.)
- Lauren Heintz (BMC Software, Inc.)
- Verne Hyde (Independent Consultant)
- Bob Natale (ACE*COMM Corporation)
- Shawn Routhier (Epilogue)
- Mike Thatcher (Independent Consultant)
- Bert Wijnen (IBM T. J. Watson Research Center)

7. Security Considerations

There is a single management object defined in this MIB that has a MAX-ACCESS clause of read-write. This object may be considered sensitive or vulnerable in some network environments. The support for SET operations in a non-secure environment without proper protection can have a negative effect on network operations.

There is a single managed object in this MIB that may contain sensitive information. This object is `agentxSessionAdminStatus`. Setting `agentxSessionAdminStatus` to an inappropriate value can effectively prevent access to management information, or provide access to inappropriate information.

It is thus important to control even GET access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. Not all versions of SNMP provide features for such a secure environment.

SNMPv1 by itself is not a secure environment. 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.

It is recommended that the implementers consider the security features as provided by the SNMPv3 framework. Specifically, the use of the User-based Security Model RFC 2574 [12] and the View-based Access Control Model RFC 2575 [15] is recommended.

It is then a customer/user 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/delete) them.

8. References

- [1] Harrington, D., Presuhn, R. and B. Wijnen, "An Architecture for Describing SNMP Management Frameworks", RFC 2571, April 1999.
- [2] Rose, M. and K. McCloghrie, "Structure and Identification of Management Information for TCP/IP-based Internets", STD 16, RFC 1155, May 1990.
- [3] Rose, M. and K. McCloghrie, "Concise MIB Definitions", STD 16, RFC 1212, March 1991.
- [4] Rose, M., "A Convention for Defining Traps for use with the SNMP", RFC 1215, March 1991.
- [5] 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.

- [6] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Textual Conventions for SMIV2", STD 58, RFC 2579, April 1999.
- [7] McCloghrie, K., Perkins, D., Schoenwaelder, J., Case, J., Rose, M. and S. Waldbusser, "Conformance Statements for SMIV2", STD 58, RFC 2580, April 1999.
- [8] Case, J., Fedor, M., Schoffstall, M. and J. Davin, "Simple Network Management Protocol", STD 15, RFC 1157, May 1990.
- [9] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Introduction to Community-based SNMPv2", RFC 1901, January 1996.
- [10] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1906, January 1996.
- [11] Case, J., Harrington D., Presuhn R. and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)", RFC 2572, April 1999.
- [12] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", RFC 2574, April 1999.
- [13] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser, "Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2)", RFC 1905, January 1996.
- [14] Levi, D., Meyer, P. and B. Stewart, "SNMP Applications", RFC 2573, April 1999.
- [15] Wijnen, B., Presuhn, R. and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", RFC 2575, April 1999.
- [16] Case, J., Mundy, R., Partain, D. and B. Stewart, "Introduction to Version 3 of the Internet-standard Network Management Framework", RFC 2570, April 1999.
- [17] Daniele, M., Wijnen, B., Ellison, M. and D. Francisco, "Agent Extensibility (AgentX) Protocol, Version 1", RFC 2741, January 2000.
- [18] Rose, M., "SNMP MUX Protocol and MIB", RFC 1227, May 1991.

[19] Case, J., McCloghrie, K., Rose, M. and S. Waldbusser,
"Management Information Base for Version 2 of the Simple Network
Management Protocol (SNMPv2)", RFC 1907, January 1996.

9. Authors' and Editor's Addresses

Lauren Heintz
Cisco Systems
1450 North McDowell Blvd.
Petaluma, CA 94954-6515
USA
Phone: +1 707-793-1714
EMail: lheintz@cisco.com

Smitha Gudur
Independent Consultant
EMail: sgudur@hotmail.com

Mark Ellison (WG editor)
Ellison Software Consulting, Inc.
38 Salem Road
Atkinso, NH 03811
USA
Phone: +1 603-362-9270
Email: ellison@world.std.com

10. Full Copyright Statement

Copyright (C) The Internet Society (2000). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Acknowledgement

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

