Definitions of Managed Objects for lock via network management protocols

draft-meng-fan-lock-mib-02

Status of This Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of BCP 78 and BCP 79. This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on April 26, 2010.

Copyright Notice

Copyright (c) 2009 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust’s Legal
Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. It describes managed objects used for monitoring locks on a device,

Table of Contents

1. Introduction ................................................. 3
2. The Internet-Standard Management Framework ................ 4
3. Conventions .................................................. 4
4. Overview ..................................................... 4
   4.1. lockTable ............................................... 5
   4.2. Specific NM interface lock Table ...................... 5
      4.2.1. lockNetconfTable .................................. 5
      4.2.2. lockCopsPrTable ................................... 6
   4.3. Statistics ............................................... 6
   4.4. MIB References ......................................... 6
   4.5. Implementation Issues .................................. 6
5. Definitions .................................................. 7
6. Security Considerations .................................... 17
7. IANA Considerations ........................................ 18
8. Acknowledgments ............................................ 18
9. References .................................................. 18
   9.1. Normative References .................................. 18
   9.2. Informative References ................................. 20
1. Introduction

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols in the Internet community. It describes managed objects used for monitoring locks on a device.

There might be multiple network management protocols supported and used at the same time by a network device. For example, a firewall box supports NETCONF configuration, but at the same time, allows SNMP to set a few security features and COPS-PR to distribute access control rules to it. NETCONF and COPS have recognized potential conflicts between write operations issued by native protocols or foreign protocols in this context. SNMP might be extended to support locking as well as NETCONF and COPS do. In addition, there might be some proprietary protocols using locking.

The Network Configuration Protocol defined in RFC4741 provides mechanisms to install, manipulate, and delete the configuration of network devices. It provides global lock mechanism via <lock> and <unlock> operations for allowing only one session at a time to make a change to a managed device. However, draft-ietf-netconf-partial-lock defines partial lock mechanism as a capability to extend base NETCONF for allowing multiple sessions to be able to modify the configuration of a managed device in parallel.

COPS usage for Policy Provisioning (COPS-PR) defined in RFC 3084 provides mechanisms and conventions used to communicate provisioned information (QoS, Security policy, etc.) between PEPs and PDPs. Between PEP and PDP, there is a single connection per Client-Type for a given area of policy (e.g., QoS) so at a given time there must be only one server updates a particular policy configuration. Such a policy configuration is effectively locked.

The primary purpose of this MIB module is to allow operators to monitor all locks supported by a managed device and understand how they might be impacting the operations of the device and the network. A customer might be using applications that manage a particular type of functionality, e.g., those applications might use SNMP to configure firewalls, but an operator applying a NETCONF lock to the firewall might prevent SNMP from configuring the firewall. This document defines a mechanism for the operator to determine why the firewall is not getting configured on a timely basis, i.e., due to cross-interface locking. From LOCK-MIB retrieval users can get the information whether or not cross-interface locking cause, let’s say, "resourceUnavailable" errorCode. If Yes, the users might not need retrying intensively in a short period.

Now this MIB module only contains managed objects for generic locks and specific NETCONF and COPS-PR locks. If a device wants to support
any other specific locks, this module can be extended to contain more appropriate objects.

Whether an implementer wants to make it possible to monitor locks from various NM interfaces via LOCK-MIB is an implementation decision, but if so, this MIB module specification provides a standardized format and subtree for the information.

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 RFC3410[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[RFC2578], STD 59[RFC2579] and STD 60[RFC2580].

3. Conventions

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 RFC2119 [RFC2119].

4. Overview

The primary purpose of this mib module is to allow operators to monitor locks and understand how they might be impacting the operations of the device and the network. With such a MIB module, SNMP could be used to monitor locks set by NETCONF, COPS-PR (or any other protocols).

It will be useful for a SNMP operator to retrieve particular information of active locks before sending SET request. If there is any lock that intersects the requested objects, the operator could know how the SET request would be impacted by locks, e.g., the SET request would be delayed or denied by locks.

A SNMP manager should not be permitted to access locked objects. If the device detects a SET request attempting to access locked objects, it should probably response an error to the operator letting him know that the SET request could not be completed because certain network management entity has locked the relevant resources. The error type should be "resource unavailable (13)"[RFC3416].
4.1. lockTable

There is generic information for all locks via any network management protocol. lockTable, which contains entries for all locks, includes generic information of locks such as the protocol by which a lock is acquired, the principal who owns a lock, and duration a lock lasts, etc. This table exists on devices that can be managed and configured via SNMP, NETCONF, COPS-PR or any other protocols.

This mib module also defines an object named lockNMInterfacesSupported to indicate which NM interface locks are monitored. For example, if there is no COPS-PR lock in lockTable, it may due to COPS-PR protocol is not supported, COPS-PR locks are not monitored or no active COPS-PR locks at all. With this indicator, an operator can understand the situation better.

4.2. Specific NM interface lock Table

Now two NM interface locks monitoring are modeled. There might be more tables for any other specific NM interface locks monitoring in the future.

This lockNetconfTable provides specific information on each lock instance set by Netconf NM interface in lockTable.

The lockCopsPrTable provides specific information on each lock instance set by COPS-PR NM interface in lockTable.

4.2.1. lockNetconfTable

The lockNetconfTable represents the specific information of locks via NETCONF, including the session ID who own a lock, the lock ID, the datastore affected (running/candidate/startup/etc.), and the scope of the lock, etc.

There is an object named lockNetconfModified for indicating the candidate datastore is modified or not. It is useful to have the flag for modified/unmodified status of the locked objects. Once locked, it remains unmodified until somebody changes the data. Once committed, it is clean again. This could be useful information for somebody considering unlock by force.

lockNetconfModified flag could also be useful for an operator to understand that a device is in the process of active maintenance (its config is likely to change soon). For example, if the lock owner is authorized to make such changes, the operator can probably ignore SNMP interface up/down traps if there is an active lock on the interface configurations. Also a security manager might take
interest when security configuration information is being modified.

### 4.2.2. lockCopsPrTable

The lockCopsPrTable represents the specific information of locks via COPS-PR, including the area of the policy provisioning identified by the client-type, policies being installed, policies being removed and policies being updated, etc.

There is an object named lockCopsPrModified for indicating the policy is modified or not. It is a useful flag as described in the lockNetconfModified object.

### 4.3. Statistics

This mib module defines 2 statistic objects to keep track of the number of active locks and failed locks, respectively.

SNMP applications could retrieve the values of these counters periodically to detect unusual events occurring. These statistic objects can help to warn about anomalous behavior.

If the number of active locks grows rapidly or there are locks lasting for a long period of time, a security problem would be encountered, e.g., somebody would be trying to launch a denial of service attack. Keeping track of the number of active locks can help to uncover this behavior.

If the number of lock failures is unusual, an attacker might be trying to lock things they are not authorized to, or somebody is attempting applying locks to different portions of datastore to see what they could get. The counter for lockFailures could help uncover this behavior.

### 4.4. MIB References

The following MIB module has IMPORTS from RFC2578, RFC2579, RFC2580, RFC3411. In REFERENCE clauses, it also refers to RFC2748, RFC4741 and [draft-ietf-netconf-partial-lock].

### 4.5. Implementation Issues

Since protocol-specific tables need insight into corresponding protocols, support for the netconf and cops-pr tables or other extended tables in the future in this MIB module would probably be easier to implement as part of the netconf or cops-pr or related implementation since they have access to the relevant...
instrumentation, and could provide that information to the SNMP agent through a master/subagent registration mechanism. The purpose of this specification is to provide a standardized format and subtree for the common information and a MIB mechanism for adding protocol-specific tables.

5. Definitions

LOCK-MIB DEFINITIONS ::= BEGIN

IMPORTS
MODULE-IDENTITY, OBJECT-TYPE,
mib-2, Unsigned32, TimeTicks,
INTEGER, BITS, IpAddress
FROM SNMPv2-SMI
RowStatus, TruthValue
FROM SNMPv2-TC
MODULE-COMPLIANCE, OBJECT-GROUP
FROM SNMPv2-CONF
SnmpAdminString
FROM SNMP-FRAMEWORK-MIB;

lockMIB MODULE-IDENTITY
LAST-UPDATED "200910230000Z"-- 23 October 2009
ORGANIZATION "IETF Operations and Management Area
Working Group (opsawg)
"

CONTACT-INFO
"Tony Meng
Postal: Huawei Symantec
3rd Floor, Section D, Keshi Building
No.28, Xinxi Rd., Shangdi, Haidian Dist.
Beijing, China 100085
Email: mengjian@huaweisymantec.com

Washam Fan
Postal: Huawei Symantec
3rd Floor, Section D, Keshi Building
No.28, Xinxi Rd., Shangdi, Haidian Dist.
Beijing, China 100085
Email: Washam.Fan@huaweisymantec.com
"

DESCRIPTION "The module defines management information for
managing locks for network management protocols.

Copyright (C) The IETF Trust (2009). This version of
this MIB module is part of RFC XXXX; see the RFC
itself for full legal notices."

-- RFC Ed.: replace XXXX with actual RFC number & remove this note
-- Administrative assignments *******************************

lockObjects     OBJECT IDENTIFIER ::= { lockMIB 1 }
lockConformance OBJECT IDENTIFIER ::= { lockMIB 2 }

-- NM interfaces supported by the agent ***********************

lockNMInterfacesSupported OBJECT-TYPE
  SYNTAX       BITS{
    Netconf(0),
    COPSPR(1)
  }
  MAX-ACCESS   read-only
  STATUS       current
  DESCRIPTION  " Network management interfaces which supporting lock operation supported by this entity which implements this lock mib module. Setting a bit to 1 indicates the specific network management interface is supported. Otherwise, it is not supported."
  ::= { lockObjects 1 }

-- Statistics for the Lock Model *******************************

lockStatistics         OBJECT IDENTIFIER ::= { lockObjects 2 }

lockActivelocks OBJECT-TYPE
  SYNTAX     Gauge32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION  "The total number of active locks."
  ::= { lockStatistics 1 }

lockFailures OBJECT-TYPE
  SYNTAX     Counter32
  MAX-ACCESS read-only
  STATUS current
  DESCRIPTION  "The total number of failed locks."
  ::= { lockStatistics 2 }

-- Basic information about the lock ***************************

lockTable OBJECT-TYPE
SYNTAX       SEQUENCE OF LockEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "Common information about locks owned by NM
interfaces. Common information that every lock
must have is presented in this table, specific
information about particular NM interface which
owing this lock is presented in a specific table.
Such as lockNetconfTable, it contains specific
information about Netconf lock."

::= { lockObjects 3 }

lockEntry OBJECT-TYPE
SYNTAX       LockEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "Common information about a lock owned by a
particular NM interface."
INDEX        {
                 lockIndex
               }
::= { lockTable 1 }

LockEntry ::= SEQUENCE
             {
                 lockIndex             Unsigned32,
                 lockUserName          SnmpAdminString,
                 lockNMInterface       SnmpAdminString,
                 lockType              INTEGER,
                 lockStartTime         TimeTicks,
                 lockEndTime           TimeTicks,
                 lockState             INTEGER
               }

lockIndex  OBJECT-TYPE
SYNTAX       Unsigned32 (1..4294967295)
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "A unique value, greater than zero, for each lock.
This value is assigned contiguously starting from 1 by
the agent.
If the system implementing this mib module is reset, it
must assign this value contiguously from the last one
it assigned."

::= { lockEntry 1 }

lockUserName  OBJECT-TYPE
SYNTAX       SnmpAdminString (SIZE (0..255))
MAX-ACCESS read-only
STATUS current
DESCRIPTION "A human readable user name identifying
the owner of the lock. If the name is not known,
it must be the empty string (‘’H’’).
It also may be an application name varies
depending on the NM interface. The max length allowed
is 255, and the value exceeding the limit MUST be
truncated.
"

 ::= { lockEntry 2 }

lockNMInterface OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The lock represented in this conceptual row is
set by Netconf, then this value is ‘lockNetconf’.
If there are any network management interfaces
defined in the future, then this value could be it.
"

 ::= { lockEntry 3 }

lockType OBJECT-TYPE
SYNTAX INTEGER {
global(1),
partial(2)
}
MAX-ACCESS read-only
STATUS current
DESCRIPTION "Represents the type of this lock, global
lock or partial lock."

 ::= { lockEntry 4 }

lockStartTime OBJECT-TYPE
SYNTAX TimeStamp
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This value equals to the time
when the lock is set. For the locks which are
valid across SNMP system reboot, the startTime should always
be set to 0 which implies a start time that
preceded the last reboot.
"

 ::= { lockEntry 5 }
lockEndTime  OBJECT-TYPE
   SYNTAX      TimeStamp
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION "This value equals to the time when the lock is unlocked."
   ::= { lockEntry 6 }

lockState  OBJECT-TYPE
   SYNTAX      INTEGER{
               ACTIVE(1),
               FAILED(2),
               DONE(3)
               }
   MAX-ACCESS  read-only
   STATUS      current
   DESCRIPTION "The current state of the lock in this lock table. The value 'Active' represents that the lock is active currently. The value 'Failed' represents that the lock request is failed. The value 'Done' represents that the lock has been unlocked."
   ::= { lockEntry 7 }

--Lock information of specific NM interface: Netconf************************

lockNetconfTable OBJECT-TYPE
   SYNTAX      SEQUENCE OF LockNetconfEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "Information about the locked objects selected by a Netconf entity."
   ::= { lockObjects 4 }

lockNetconfEntry OBJECT-TYPE
   SYNTAX      LockNetconfEntry
   MAX-ACCESS  not-accessible
   STATUS      current
   DESCRIPTION "Information about the locked objects selected by a Netconf entity."
   INDEX       { lockIndex
              }
   ::= { lockNetconfTable 1 }
LockNetconfEntry ::= SEQUENCE

   {
      lockNetconfSessionID   Unsigned32,
      lockNetconfLockID      Unsigned32,
      lockNetconfTarget      BITS,
      lockNetconfSelectType  INTEGER,
      lockNetconfSelect      SnmpAdminString,
      lockNetconfModified    TruthValue,
      lockNetconfReleasedBy  Unsigned32
   }

lockNetconfSessionID OBJECT-TYPE
SYNTAX       Unsigned32(1..4294967295)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The sessionID of the Netconf session
 which owns this lock."
::= { lockNetconfEntry 2 }

lockNetconfLockID OBJECT-TYPE
SYNTAX       Unsigned32
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "This value is set to the Netconf lockID of this lock."
::= { lockNetconfEntry 3 }

lockNetconfTarget OBJECT-TYPE
SYNTAX       BITS{
             Running(0),
             Candidate(1)
        }
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "Represents the target of this lock."
::= { lockNetconfEntry 4 }

lockNetconfSelectType OBJECT-TYPE
SYNTAX       INTEGER{
             XPath(1),
             Subtree(2)
        }
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "Represents the type of expression, XPath or subtree."
::= { lockNetconfEntry 5 }

lockNetconfSelect OBJECT-TYPE
SYNTAX       SnmpAdminString
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "XPath or subtree expressions represent the scope of 
this lock."
::= { lockNetconfEntry 6 }

lockNetconfModified OBJECT-TYPE
SYNTAX       TruthValue
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "The value 'true' indicates that the locked 
portion of the target has been modified by the user. 
The value 'false' indicates that no modifications 
have been done yet."
::= { lockNetconfEntry 7 }

lockNetconfUnlockedBy OBJECT-TYPE
SYNTAX       Unsigned32(0..4294967295)
MAX-ACCESS   read-only
STATUS       current
DESCRIPTION  "Represents the session ID of the session 
which use unlock operation or end this lock forcibly. 
If the session which released this lock is the session 
which set this lock, then the value this instance is 0."
::= { lockNetconfEntry 8 }

--Lock information of specific NM interface: COPS-PR**************

lockCopsPrTable OBJECT-TYPE
SYNTAX       SEQUENCE OF LockCopsPrEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "Information about the locked objects 
selected by a COPS-PR entity."
::= { lockObjects 5 }

lockCopsPrEntry OBJECT-TYPE
SYNTAX       LockCopsPrEntry
MAX-ACCESS   not-accessible
STATUS       current
DESCRIPTION  "Information about the locked objects 
selected by a COPS-PR entity."
INDEX        {
  lockIndex
}
::= { lockCopsPrTable 1 }
LockCopsPrEntry ::= SEQUENCE
{
  lockCopsPrIndex          Unsigned32,
  lockCopsPrPEPID          OCTET STRING,
  lockCopsPrPDPAddr        IpAddress,
  lockCopsPrClientState    INTEGER,
  lockCopsPrClientHandle   Unsigned32,
  lockCopsPrClientType     INTEGER,
  lockCopsPrInstallPolicies SnmpAdminString,
  lockCopsPrRemovePolicies SnmpAdminString,
  lockCopsPrUpdatePolicies SnmpAdminString,
  lockCopsPrModified       TruthValue
}

lockCopsPrIndex OBJECT-TYPE
SYNTAX Unsigned32 (1..4294967295)
MAX-ACCESS not-accessible
STATUS current
DESCRIPTION "An index value that uniquely identifies lock
owned by a COPS-PR client. The lock identifier
is unique within the COPS-PR scope."
 ::= { lockCopsPrEntry 1 }

lockCopsPrPEPID OBJECT-TYPE
SYNTAX OCTET STRING
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It’s value uniquely identifies the PEP
whose policy configuration is locked. It’s value is
same as the value of PEPID defined in [RFC 2748]."
 ::= { lockCopsPrEntry 2 }

locKCopsPrPDPAddr OBJECT-TYPE
SYNTAX IpAddress
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It’s the IP address of the PDP which make
the policy decision to the PEP."
 ::= { lockCopsPrEntry 3 }

lockCopsPrClientState OBJECT-TYPE
SYNTAX INTEGER{
  CLOSE(0),
  OPEN(1)
}
MAX-ACCESS read-only
STATUS current

Meng & Fan Expires April 26, 2010 [Page 14]
DESCRIPTION "This value indicates whether a particular type
of client is supported both by the PEP identified by
PEPID and the PDP identified by LocKcopsPrPDPAddr." 
::= { lockCopsPrEntry 4 }

lockCopsPrClientType OBJECT-TYPE
SYNTAX INTEGER (0..65535)
MAX-ACCESS read-only
STATUS current
DESCRIPTION "This value uniquely identifies the area of policy
configuration."
::= { lockCopsPrEntry 5 }

lockCopsPrClientHandle OBJECT-TYPE
SYNTAX Unsigned32
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The client handle value is used to uniquely
identify a particular PEP’s request among other
currently installed requests."
::= { lockCopsPrEntry 6 }

lockCopsPrInstallPolicies OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It’s value indicates the provisioned policies to be
installed by the PEP."
::= { lockCopsPrEntry 7 }

lockCopsPrRemovePolicies OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It’s value indicates the provisioned policies to be
deleted by the PEP."
::= { lockCopsPrEntry 8 }

lockCopsPrUpdatePolicies OBJECT-TYPE
SYNTAX SnmpAdminString
MAX-ACCESS read-only
STATUS current
DESCRIPTION "It’s value indicates the provisioned policies to be
updated by the PEP."
::= { lockCopsPrEntry 9 }

lockCopsPrModified OBJECT-TYPE
SYNTAX TruthValue
MAX-ACCESS read-only
STATUS current
DESCRIPTION "The value ‘true’ indicates that the locked portion of
the target has been modified by the user.
The value ‘false’ indicates that no modifications have
been done yet."
 ::= { lockCopsPrEntry 10 }

-- Conformance Information *******************************************

lockCompliances OBJECT IDENTIFIER ::= { lockConformance 1 }
lockGroups OBJECT IDENTIFIER ::= { lockConformance 2 }

-- Compliance statements

lockCompliance MODULE-COMPLIANCE
 STATUS current
 DESCRIPTION "The compliance statement for an entity who implements
this LOCK-MIB."

MODULE -- this module
MANDATORY-GROUPS { lockBasicGroup }

GROUP lockNetconfGroup
 DESCRIPTION "This group is mandatory only for those
entities which implement Netconf."

GROUP lockCopsPrGroup
 DESCRIPTION "This group is mandatory only for those
entities which implement COPS-PR."

 ::= { lockCompliances 1 }

lockBasicGroup OBJECT-GROUP
 OBJECTS { lockNMInterfacesSupported, lockUserName, lockNMInterface, lockType, lockStartTime, lockEndTime, lockState }
 STATUS current
 DESCRIPTION "A collection of objects providing basic
instrumentation of an entity which implements
the lock managing and monitoring."
 ::= { lockGroups 1 }
lockNetconfGroup OBJECT-GROUP
   OBJECTS     {
      lockNetconfSessionID,
      lockNetconfLockID,
      lockNetconfTarget,
      lockNetconfSelectType,
      lockNetconfSelect,
      lockNetconfModified,
      lockNetconfUnlockedBy
   }
   STATUS       current
   DESCRIPTION "A collection of objects providing basic instrumentation
of an entity which supports Netconf."
   ::= { lockGroups 2 }

lockCopsPrGroup OBJECT-GROUP
   OBJECTS     {
      lockCopsPrPEPID,
      lockCopsPrPDPAddr,
      lockcopsPrClientState,
      lockCopsPrClientHandle,
      lockCopsPrClientType,
      lockCopsPrInstallPolicies,
      lockCopsPrRemovePolicies,
      lockCopsPrUpdatePolicies,
      lockCopsPrModified
   }
   STATUS       current
   DESCRIPTION "A collection of objects providing basic instrumentation
of an entity which supports COPS-PR."
   ::= { lockGroups 3 }

END

6. Security Considerations

There are no management objects defined in this MIB module that have
a MAX-ACCESS clause of read-write and/or read-create. So, if this
MIB module is implemented correctly, then there is no risk that an
intruder can alter or create any management objects of this MIB
module via direct SNMP SET operations.

The readable objects in this MIB module are not sensitive.
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.

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

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

7. IANA Considerations

The MIB module in this document uses the following IANA-assigned OBJECT IDENTIFIER values recorded in the SMI Numbers registry:

<table>
<thead>
<tr>
<th>Descriptor</th>
<th>OBJECT IDENTIFIER value</th>
</tr>
</thead>
<tbody>
<tr>
<td>lockMIB</td>
<td>{ mib-2 XXX }</td>
</tr>
</tbody>
</table>

Editor’s Note (to be removed prior to publication): the IANA is requested to assign a value for "XXX" under the ‘mib-2′ subtree and to record the assignment in the SMI Numbers registry.

8. Acknowledgments

Many thanks to David Harrington for his guidance and feedback on this MIB module.

9. References

9.1. Normative References


9.2.  Informative References


Authors’ Addresses

Tony Meng
Huawei Symantec
3rd Floor, Section D, Keshi Building
No.28, Xinxi Rd., Shangdi, Haidian Dist.
Beijing 100085
China
EMail: mengjian@huaweisymantec.com
URI: http://www.huaweisymantec.com

Washam Fan
Huawei Symantec
3rd Floor, Section D, Keshi Building
No.28, Xinxi Rd., Shangdi, Haidian Dist.
Beijing 100085
China
EMail: Washam.Fan@huaweisymantec.com
URI: http://www.huaweisymantec.com