

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: March 18, 2011

K. Narayan
Cisco Systems, Inc.
D. Nelson
Elbrys Networks, Inc.
R. Presuhn, Ed.
None
September 14, 2010

Using Authentication, Authorization, and Accounting services to
Dynamically Provision View-based Access Control Model User-to-Group
Mappings
draft-ietf-isms-radius-vacm-11.txt

Abstract

This memo defines a portion of the Management Information Base (MIB) for use with network management protocols. It describes the use of information provided by Authentication, Authorization, and Accounting (AAA) services, such as the Remote Authentication Dial-In User Service (RADIUS), to dynamically update user-to-group mappings in the View-Based Access Control Model (VACM).

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on March 18, 2011.

Copyright Notice

Copyright (c) 2010 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 Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	3
2. The Internet-Standard Management Framework	3
3. Conventions	3
4. Overview	4
4.1. Using AAA services with SNMP	4
4.2. Applicability	5
5. Structure of the MIB Module	6
5.1. Textual Conventions	6
5.2. The Table Structure	6
6. Relationship to Other MIB Modules	6
6.1. Relationship to the VACM MIB	6
6.2. MIB modules required for IMPORTS	6
6.3. Documents required for REFERENCE clauses	6
7. Elements of Procedure	7
7.1. Sequencing Requirements	7
7.2. Actions Upon Session Establishment Indication	7
7.2.1. Required Information	7
7.2.2. Creation of Entries in vacmAaaSecurityToGroupTable	8
7.2.3. Creation of Entries in vacmSecurityToGroupTable	8
7.2.4. Update of vacmGroupName	9
7.3. Actions Upon Session Termination Indication	9
7.3.1. Deletion of Entries from vacmAaaSecurityToGroupTable	9
7.3.2. Deletion of Entries from vacmSecurityToGroupTable	10
8. Definitions	10
9. Security Considerations	14
9.1. Principal Identity Naming	14
9.2. Management Information Considerations	15
10. IANA Considerations	16
11. Contributors	17
12. References	18
12.1. Normative References	18
12.2. Informative References	19
Authors' Addresses	19

1. Introduction

This memo specifies a way to dynamically provision selected View-Based Access Control Model (VACM) [RFC3415] Management Information Base (MIB) objects, based on information received from an Authentication, Authorization, and Accounting (AAA) service, such as RADIUS [RFC2865] and [RFC5607]. It reduces the need for security administrators to manually update VACM configurations due to user churn, allowing a centralized AAA service to provide the information associating a given user with the access control policy (known as a "group" in VACM) governing that user's access to management information.

This memo requires no changes to the Abstract Service Interface for the Access Control Subsystem, and requires no changes to the Elements of Procedure for VACM. It provides a MIB module that reflects the information provided by the AAA service, along with elements of procedure for maintaining that information and performing corresponding updates to VACM MIB data.

The reader is expected to be familiar with [RFC3415], [RFC5607], [RFC5608], and their supporting specifications.

2. The Internet-Standard Management Framework

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

Managed objects are accessed via a virtual information store, termed the Management Information Base or MIB. MIB objects are generally accessed through the Simple Network Management Protocol (SNMP). Objects in the MIB are defined using the mechanisms defined in the Structure of Management Information (SMI). This memo specifies a MIB module that is compliant to the SMIv2, which is described in STD 58, RFC 2578 [RFC2578], STD 58, RFC 2579 [RFC2579] and STD 58, RFC 2580 [RFC2580].

3. Conventions

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

4. Overview

4.1. Using AAA services with SNMP

There are two use cases for AAA support of management access via SNMP. These are (a) service authorization and (b) access control authorization. The former is discussed in detail in [RFC5608]. The latter is the subject of this memo.

The use case assumption here is that roles within an organization, which are reflected in VACM as groups, naming access control policies, change infrequently, while the users assigned to those roles change much more frequently. This memo describes how the user-to-role (group) mapping can be delegated to the RADIUS server, avoiding the need to re-provision managed devices as users are added, deleted, or assigned new roles in an organization.

This memo assumes that the detailed access control policies are pre-configured in VACM, and does not attempt to address the question of how the policy associated with a given role is put in place.

The only additional information obtained from the AAA service is the mapping of the authenticated user's identifier to a specific role (or "group" in VACM terminology) in the access control policy. Dynamic user authorization for MIB database access control, as defined herein, is limited to mapping the authenticated user to a group, which in turn is mapped to whatever access control policies are already in place in VACM.

The SNMP architecture [RFC3411] maintains strong modularity and separation of concerns, separating user identity (authentication) from user database access rights (authorization). RADIUS, on the other hand, allows for no such separation of authorization from authentication. Consequently, the approach here is to leverage existing RADIUS usage for identifying a principal, documented in [RFC5608], along with the RADIUS Management-Policy-Id Attribute [RFC5607].

A unique identifier is needed for each AAA-authorized "session", corresponding to a communication channel, such as a transport session, for which a principal has been AAA-authenticated and which is authorized to offer SNMP service. How these identifiers are assigned is implementation-dependent. When a RADIUS Management-Policy-Id Attribute (or equivalent) is bound to such a session and principal authentication, this binding provides sufficient information to compute dynamic updates to VACM. How this information is communicated within an implementation is implementation dependent; this memo is only concerned with externally observable behavior.

The key concept here is that what we will informally call a "AAA binding" binds:

1. a communications channel
2. an authenticated principal
3. service authorization
4. an access control policy name

Some of the binding is done via other specifications. A transport model, such as the Secure Shell Transport Model [RFC5592], provides a binding between 1) and 2) and 3), providing a securityName. In turn, [RFC5607] provides a binding between (1+2+3) and 4). This document extends that further, to create a binding between (1+2+3+4) and the local (VACM MIB) definition of the named policy, called a group in VACM.

4.2. Applicability

Though this memo was motivated to support the use of specific Transport Models, such as the Secure Shell Transport Model [RFC5592], it MAY be used with other implementation environments satisfying these requirements:

- o use an AAA service for sign-on service and data access authorization;
- o provide an indication of the start of a session for a particular authenticated principal, identified using an SNMP securityName [RFC3411], and provide the corresponding value to be used to identify a VACM group to be used, based on information provided by the AAA service in use;
- o provide an indication of the end of the need for being able to make access decisions for a particular authenticated principal, as at the end of a session, whether due to disconnection, termination due to timeout, or any other reason.

Likewise, although this memo specifically refers to RADIUS, it MAY be used with other AAA services satisfying these requirements:

- o the service provides information semantically equivalent to the RADIUS Management-Policy-Id Attribute [RFC5607], which corresponds to the name of a VACM group;

- o the service provides an authenticated principal identifier (e.g., the RADIUS User-Name Attribute [RFC2865]) which can be transformed to an equivalent principal identifier in the form of a securityName [RFC3411].

5. Structure of the MIB Module

5.1. Textual Conventions

This MIB module makes use of the SnmpAdminString [RFC3411] and SnmpSecurityModel [RFC3411] textual conventions.

5.2. The Table Structure

This MIB module defines a single table, the vacmAaaSecurityToGroupTable. This table is indexed by the integer assigned to each security model, the protocol-independent securityName corresponding to a principal, and the unique identifier of a session.

6. Relationship to Other MIB Modules

This MIB module has a close operational relationship with the SNMP-VIEW-BASED-ACM-MIB (more commonly known as the "VACM MIB") from [RFC3415]. It also relies on IMPORTS from several other modules.

6.1. Relationship to the VACM MIB

Although the MIB module defined here has a close relationship with the VACM MIB's vacmSecurityToGroupTable, it in no way changes the elements of procedure for VACM, nor does it affect any other tables defined in VACM. See the elements of procedure (below) for details of how the contents of the vacmSecurityToGroupTable are affected by this MIB module.

6.2. MIB modules required for IMPORTS

This MIB module employs definitions from [RFC2578], [RFC2579] and [RFC3411].

6.3. Documents required for REFERENCE clauses

This MIB module contains REFERENCE clauses making reference to [RFC2865], [RFC3411], and [RFC5590].

7. Elements of Procedure

The following elements of procedure are formulated in terms of two types of events: an indication of the establishment of a session, and an indication that one has ended. These can result in the creation of entries in the `vacmAaaSecurityToGroupTable`, which can in turn trigger creation, update, or deletion of entries in the `vacmSecurityToGroupTable`.

There are various possible implementation-dependent error cases not spelled out here, such as running out of memory. By their nature, recovery in such cases will be implementation-dependent. Implementors are advised to consider fail-safe strategies, e.g., prematurely terminating access in preference to erroneously perpetuating access.

7.1. Sequencing Requirements

These procedures assume that a transport model, such as [RFC5592], coordinates session establishment with AAA authentication and authorization. They rely on the receipt by the AAA client of the RADIUS Management-Policy-Id [RFC5607] Attribute (or its equivalent) from the RADIUS Access-Accept message (or equivalent). They also assume that the User-Name [RFC2865] from the RADIUS Access-Request message (or equivalent) corresponds to a `securityName` [RFC3411].

To ensure correct processing of SNMP PDUs, the handling of the indication of the establishment of a session in accordance with the elements of procedure below MUST be completed before the `isAccessAllowed()` abstract service interface [RFC3415] is invoked for any SNMP PDUs from that session.

If a session termination indication occurs before all invocations of the `isAccessAllowed()` abstract service interface [RFC3415] have completed for all SNMP PDUs from that session, those remaining invocations MAY result in denial of access.

7.2. Actions Upon Session Establishment Indication

7.2.1. Required Information

Four pieces of information are needed to process the session establishment indication:

- o the `SnmpSecurityModel` [RFC3411] needed as an index into the `vacmSecurityToGroupTable`;

- o the RADIUS User-Name Attribute;
- o a session identifier, as a unique, definitive identifier of the session that the AAA authorization is tied to;
- o the RADIUS Management-Policy-Id Attribute.

All four of these pieces of information are REQUIRED. In particular, if either the User-Name or Management-Policy-Id is absent, invalid, or a zero-length string, no further processing of the session establishment indication is undertaken.

As noted in Section 4.2, the above text refers specifically to RADIUS attributes. Other AAA services can be substituted, but the requirements imposed on the User-Name and the Management-Policy-Id-Attribute MUST be satisfied using the equivalent fields for those services.

7.2.2. Creation of Entries in vacmAaaSecurityToGroupTable

Whenever an indication arrives that a new session has been established, determine whether a corresponding entry exists in the vacmAaaSecurityToGroupTable. If one does not, create a new row with the columns populated as follows:

- o vacmAaaSecurityModel = value of SnmpSecurityModel corresponding to the security model in use;
- o vacmAaaSecurityName = RADIUS User-Name Attribute or equivalent, the securityName that will be used in invocations of the isAccessAllowed() abstract service interface [RFC3415];
- o vacmAaaSessionID = session identifier, unique across all open sessions of all of this SNMP engine's transport models;
- o vacmAaaGroupName = RADIUS Management-Policy-Id Attribute or equivalent.

Otherwise, if the row already exists, update the vacmAaaGroupName with the the RADIUS Management-Policy-Id Attribute or equivalent supplied.

7.2.3. Creation of Entries in vacmSecurityToGroupTable

Whenever an entry is created in the vacmAaaSecurityToGroupTable, the vacmSecurityToGroupTable is examined to determine whether a corresponding entry exists there, using the value of vacmAaaSecurityModel for vacmSecurityModel, and the value of

vacmAaaSecurityName for vacmSecurityName. If no corresponding entry exists, create one, using the vacmAaaGroupName of the newly created entry to fill in vacmGroupName, using a value of "volatile" for the row's StorageType, and a value of "active" for its RowStatus.

7.2.4. Update of vacmGroupName

Whenever the value of an instance of vacmAaaGroupName is updated, if a corresponding entry exists in the vacmSecurityToGroupTable, and that entry's StorageType is "volatile" and its RowStatus is "active", update the value of vacmGroupName with the value from vacmAaaGroupName.

If a corresponding entry already exists in the vacmSecurityToGroupTable, and that row's StorageType is anything other than "volatile", or its RowStatus is anything other than "active", then that instance of vacmGroupName MUST NOT be modified.

The operational assumption here is that if the row's StorageType is "volatile", then this entry was probably dynamically created; an entry created by a security administrator would not normally be given a StorageType of "volatile". If value being provided by RADIUS (or other AAA service) is the same as what is already there, this is a no-op. If the value is different, the new information is understood as a more recent role (group) assignment for the user, which should supersede the one currently held there. The structure of the vacmSecurityToGroupTable makes it impossible for a (vacmSecurityModel, vacmSecurityName) tuple to map to more than one group.

7.3. Actions Upon Session Termination Indication

Whenever a RADIUS (or other AAA) authenticated session ends for any reason, an indication is provided. This indication MUST provide means of determining the SnmpSecurityModel, and an identifier for the transport session tied to the AAA authorization. The manner in which this occurs is implementation dependent.

7.3.1. Deletion of Entries from vacmAaaSecurityToGroupTable

Entries in the vacmAaaSecurityToGroupTable MUST NOT persist across system reboots.

When a session has been terminated, the vacmAaaSecurityToGroupTable is searched for a corresponding entry. A "matching" entry is any entry for which the SnmpSecurityModel and session ID match the information associated with the session termination indication. Any matching entries are deleted. It is possible that no entries will

match; this is not an error, and no special processing is required in this case.

7.3.2. Deletion of Entries from vacmSecurityToGroupTable

Whenever the last remaining row bearing a particular (vacmAaaSecurityModel, vacmAaaSecurityName) pair is deleted from the vacmAaaSecurityToGroupTable, the vacmSecurityToGroupTable is examined for a corresponding row. If one exists, and if its StorageType is "volatile" and its RowStatus is "active", that row MUST be deleted as well. The mechanism to accomplish this task is implementation-dependent.

8. Definitions

SNMP-VACM-AAA-MIB DEFINITIONS ::= BEGIN

IMPORTS

```
MODULE-COMPLIANCE, OBJECT-GROUP          FROM SNMPv2-CONF
MODULE-IDENTITY, OBJECT-TYPE,
mib-2,
Unsigned32                                FROM SNMPv2-SMI
SnmpAdminString,
SnmpSecurityModel                        FROM SNMP-FRAMEWORK-MIB;
```

vacmAaaMIB MODULE-IDENTITY

```
LAST-UPDATED "201009010000Z"              -- 1 September, 2010
ORGANIZATION "ISMS Working Group"
CONTACT-INFO "WG-email: isms@ietf.org"
```

```
DESCRIPTION "The management and local datastore information
definitions for the AAA-Enabled View-based Access
Control Model for SNMP.
```

Copyright (c) 2010 IETF Trust and the persons
identified as the document authors. All rights
reserved.

Redistribution and use in source and binary forms,
with or without modification, is permitted pursuant
to, and subject to the license terms contained in,
the Simplified BSD License set forth in Section
4.c of the IETF Trust's Legal Provisions Relating
to IETF Documents
(<http://trustee.ietf.org/license-info>).

This version of this MIB module is part of RFC XXXX;
see the RFC itself for full legal notices."

```
REVISION "201009010000Z"
DESCRIPTION "Initial version, published as RFC XXXX."
 ::= { mib-2 XXX }
-- RFC Ed.: replace XXX with IANA-assigned number & remove this note
-- RFC Ed.: replace XXXX with the RFC number & remove this note

vacmAaaMIBObjects OBJECT IDENTIFIER ::= { vacmAaaMIB 1 }

vacmAaaMIBConformance OBJECT IDENTIFIER ::= { vacmAaaMIB 2 }

vacmAaaSecurityToGroupTable OBJECT-TYPE
    SYNTAX      SEQUENCE OF VacmAaaSecurityToGroupEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "This table provides a listing of all currently active
                  sessions for which a mapping of the combination of
                  SnmpSecurityModel and securityName into the name of
                  a VACM group which has been provided by an AAA service.
                  The group name (in VACM) in turn identifies an access
                  control policy to be used for the corresponding
                  principals."
    REFERENCE    "RFC 3411 section 3.2.2 defines securityName"
    ::= { vacmAaaMIBObjects 1 }

vacmAaaSecurityToGroupEntry OBJECT-TYPE
    SYNTAX      VacmAaaSecurityToGroupEntry
    MAX-ACCESS   not-accessible
    STATUS       current
    DESCRIPTION  "An entry in this table maps the combination of a
                  SnmpSecurityModel and securityName into the name
                  of a VACM group defining the access control policy
                  which is to govern a particular session.

                  Each entry corresponds to a session.

                  Entries do not persist across reboots.

                  An entry is created whenever an indication occurs
                  that a new session has been established that would
                  not have the same index values as an existing entry.

                  When a session is torn down, disconnected, timed out
                  (e.g., following the RADIUS Session-Timeout Attribute),
                  or otherwise terminated for any reason, the
                  corresponding vacmAaaSecurityToGroupEntry is deleted."
```

```

REFERENCE    "RFC 3411 section 3.2.2 defines securityName"
INDEX        {
              vacmAaaSecurityModel,
              vacmAaaSecurityName,
              vacmAaaSessionID
            }
::= { vacmAaaSecurityToGroupTable 1 }

```

```

VacmAaaSecurityToGroupEntry ::= SEQUENCE
{
    vacmAaaSecurityModel          SnmpSecurityModel,
    vacmAaaSecurityName           SnmpAdminString,
    vacmAaaSessionID              Unsigned32,
    vacmAaaGroupName              SnmpAdminString
}

```

```

vacmAaaSecurityModel OBJECT-TYPE
    SYNTAX      SnmpSecurityModel(1..2147483647)
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The security model associated with the AAA binding
                represented by this entry.

                This object cannot take the 'any' (0) value."
    ::= { vacmAaaSecurityToGroupEntry 1 }

```

```

vacmAaaSecurityName OBJECT-TYPE
    SYNTAX      SnmpAdminString (SIZE(1..32))
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "The securityName of the principal associated with the
                AAA binding represented by this entry.  In RADIUS
                environments, this corresponds to the User-Name
                Attribute."
    REFERENCE   "RFC 3411 section 3.2.2 defines securityName, and
                RFC 2865 section 5.1 defines User-Name."
    ::= { vacmAaaSecurityToGroupEntry 2 }

```

```

vacmAaaSessionID OBJECT-TYPE
    SYNTAX      Unsigned32
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION "An implementation-dependent identifier of the session.

                This value MUST be unique among all currently open
                sessions of all of this SNMP engine's transport models.
                The value has no particular significance other than to
                distinguish sessions.

```

Implementations in which tmSessionID has a compatible syntax and is unique across all transport models MAY use that value."

REFERENCE "The abstract service interface parameter tmSessionID is defined in RFC 5590 section 5.2.4."

::= { vacmAaaSecurityToGroupEntry 3 }

vacmAaaGroupName OBJECT-TYPE

SYNTAX SnmpAdminString (SIZE(1..32))

MAX-ACCESS read-only

STATUS current

DESCRIPTION "The name of the group to which this entry is to belong. In RADIUS environments this comes from the RADIUS Management-Policy-Id Attribute."

When the appropriate conditions are met, the value of this object is applied the vacmGroupName in the corresponding vacmSecurityToGroupEntry."

REFERENCE "RFC 3415"

::= { vacmAaaSecurityToGroupEntry 4 }

-- Conformance information *****

vacmAaaMIBCompliances

OBJECT IDENTIFIER ::= {vacmAaaMIBConformance 1}

vacmAaaMIBGroups

OBJECT IDENTIFIER ::= {vacmAaaMIBConformance 2}

-- compliance statements

vacmAaaMIBBasicCompliance MODULE-COMPLIANCE

STATUS current

DESCRIPTION "The compliance statement for SNMP engines implementing the AAA-Enabled View-based Access Control Model for SNMP."

MODULE -- this module

MANDATORY-GROUPS { vacmAaaGroup }

::= { vacmAaaMIBCompliances 1 }

-- units of conformance

vacmAaaGroup OBJECT-GROUP

OBJECTS {

vacmAaaGroupName

}

STATUS current

DESCRIPTION "A collection of objects for supporting the use of AAA services to provide user / group mappings for VACM."
 ::= { vacmAaaMIBGroups 1 }

END

9. Security Considerations

The algorithms in this memo make heuristic use of the `StorageType` of entries in the `vacmSecurityToGroupTable` to distinguish those provisioned by a security administrator (which would presumably not be configured as "volatile") from those dynamically generated. In making this distinction, it assumes that those entries explicitly provisioned by a security administrator and given a non-"volatile" status are not to be dynamically overridden. Furthermore, it assumes that any active entries with "volatile" status can be treated as dynamic, and deleted or updated as needed. Users of this memo need to be aware of this operational assumption, which, while reasonable, is not necessarily universally valid. For example, this situation could also occur if the SNMP security administrator had mistakenly created these non-volatile entries in error.

The design of VACM ensures that if an unknown policy (group name) is used in the `vacmSecurityToGroupTable`, no access is granted. A consequence of this is that no matter what information is provided by the AAA server, no user can gain SNMP access rights not already granted to some group through the VACM configuration.

9.1. Principal Identity Naming

In order to ensure that the access control policy ultimately applied as a result of the mechanisms described here is indeed the intended policy for a given principal using a particular security model, care needs to be applied in the mapping of the authenticated user (principal) identity to the `securityName` used to make the access control decision. Broadly speaking, there are two approaches to ensure consistency of identity:

- o Entries for the `vacmSecurityToGroupTable` corresponding to a given security model are created only through the operation of the procedures described in this memo. A consequence of this would be that all such entries would have been created using the RADIUS `User-Name` (or other AAA-authenticated identity) and RADIUS `Management-Policy-Id` Attribute (or equivalent).

- o Administrative policy allows a matching pre-configured entry to exist in the vacmSecurityToGroupTable, i.e., an entry with the corresponding vacmSecurityModel and with a vacmSecurityName matching the authenticated principal's RADIUS User-Name. In this case, administrative policy also needs to ensure consistency of identity between each authenticated principal's RADIUS User-Name and the administratively configured vacmSecurityName in the vacmSecurityToGroupTable row entries for that particular security model.

In the later case, inconsistent re-use of the same name for different entities or individuals (principals) can cause the incorrect access control policy to be applied for the authenticated principal, depending on whether the policy configured using SNMP, or the policy applied using the procedures of this memo, is the intended policy. This may result in greater or lesser access rights than the administrative policy intended. Inadvertent mis-identification in such cases may be undetectable by the SNMP engine or other software elements of the managed entity.

9.2. Management Information 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.

Some of the readable objects in this MIB module (including some objects with a MAX-ACCESS of not-accessible, whose values are exposed as a result access to indexed objects) may be considered sensitive or vulnerable in some network environments. It is thus important to control even GET and/or NOTIFY access to these objects and possibly to even encrypt the values of these objects when sending them over the network via SNMP. These are the tables and objects and their sensitivity/vulnerability:

- o vacmAaaSecurityToGroupTable - the entire table is potentially sensitive, since walking the table will reveal user names, security models in use, session identifiers, and group names;
- o vacmAaaSecurityModel - though not-accessible, this is exposed as an index of vacmAaaGroupName;
- o vacmAaaSecurityName - though not-accessible, this is exposed as an index of vacmAaaGroupName;

- o vacmAaaSessionID - though not-accessible, this is exposed as an index of vacmAaaGroupName;
- o vacmAaaGroupName - since this identifies a security policy and associates it with a particular user, this is potentially 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.

10. IANA Considerations

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

Descriptor	OBJECT IDENTIFIER value
-----	-----
vacmAaaMIB	{ mib-2 XXX }

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. When the assignment has been made, the RFC Editor is asked to replace "XXX" (here and in the MIB module) with the assigned value and to remove this note.

11. Contributors

The following participants from the isms working group contributed to the development of this document:

- o Andrew Donati
- o David Harrington
- o Jeffrey Hutzelman
- o Juergen Schoenwaelder
- o Tom Petch
- o Wes Hardaker

During the IESG review additional comments were received from:

- o Adrian Farrel
- o Amanda Baber
- o Dan Romescanu
- o David Kessens
- o Francis Dupont
- o Glenn Keeni
- o Jari Arkko
- o Joel Jaeggli
- o Magnus Nystroem
- o Mike Heard
- o Robert Story
- o Russ Housley
- o Sean Turner
- o Tim Polk

12. References

12.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC2578] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Structure of Management Information Version 2 (SMIv2)", STD 58, RFC 2578, April 1999.
- [RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.
- [RFC2580] McCloghrie, K., Perkins, D., and J. Schoenwaelder, "Conformance Statements for SMIv2", STD 58, RFC 2580, April 1999.
- [RFC2865] Rigney, C., Willens, S., Rubens, A., and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000.
- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.
- [RFC5590] Harrington, D. and J. Schoenwaelder, "Transport Subsystem for the Simple Network Management Protocol (SNMP)", RFC 5590, June 2009.
- [RFC5607] Nelson, D. and G. Weber, "Remote Authentication Dial-In User Service (RADIUS) Authorization for Network Access Server (NAS) Management", RFC 5607, July 2009.
- [RFC5608] Narayan, K. and D. Nelson, "Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models", RFC 5608, August 2009.

12.2. Informative References

- [RFC3410] Case, J., Mundy, R., Partain, D., and B. Stewart,
"Introduction and Applicability Statements for Internet-
Standard Management Framework", RFC 3410, December 2002.
- [RFC5592] Harrington, D., Salowey, J., and W. Hardaker, "Secure
Shell Transport Model for the Simple Network Management
Protocol (SNMP)", RFC 5592, June 2009.

Authors' Addresses

Kaushik Narayan
Cisco Systems, Inc.
10 West Tasman Drive
San Jose, CA 95134
USA

Phone: +1 408-526-8168
Email: kaushik_narayan@yahoo.com

David Nelson
Elbrys Networks, Inc.
282 Corporate Drive, Unit #1,
Portsmouth, NH 03801
USA

Phone: +1 603-570-2636
Email: d.b.nelson@comcast.net

Randy Presuhn (editor)
None
San Jose, CA 95120
USA

Email: randy_presuhn@mindspring.com

isms
Internet-Draft
Intended status: Standards Track
Expires: April 28, 2011

R. Pejaver
Y. Lee
Comcast
W. Hardaker
SPARTA, Inc.
K. Hornstein
US Naval Research Laboratory
October 25, 2010

Kerberos Security Model for SNMPv3
draft-pejaver-isms-kerberos-01

Abstract

This memo describes the use of Kerberos service with Simple Network Management Protocol (SNMP) to authenticate users, authorize them to access specific MIB objects, and to protect SNMP messages by using integrity protection and encryption. User authorization information is securely encapsulated inside the AuthorizationData field in a Kerberos ticket.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on April 28, 2011.

Copyright Notice

Copyright (c) 2010 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 Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Requirements Language	3
2. Introduction	3
2.1. General	3
2.2. Use Cases	3
2.3. Security Model Requirements	4
2.4. Message Protection Requirements	5
2.5. Kerberos Service Operational Model	5
2.6. User Authorization Information	6
3. Elements of the model	6
3.1. How KSM fits into the SNMP Architecture	7
3.2. High Level Architecture	9
3.3. Outline of the Proposal	10
3.4. Services for generating an outgoing SNMP request	10
3.5. Services for generating an outgoing SNMP response	11
3.6. Services for processing an incoming SNMP message	12
4. Elements of the procedure	13
4.1. Procedure for outgoing requests	13
4.2. Procedure for incoming requests	14
4.3. Procedure for authorizing incoming requests	15
4.4. Procedure for outgoing responses	15
4.5. Procedure for incoming responses	16
5. Unconfirmed messages	16
6. Comparative Analysis	16
6.1. Addressing the requirements	16
6.2. Advantages of KSM	16
6.3. Disadvantages of this Security Model	17
6.4. Issues with models based on transport sessions	17
7. IANA Considerations	18
8. Security Considerations	18
9. Acknowledgements	18
10. References	18
10.1. Normative References	18
10.2. Informative References	19
Appendix A. Implementation Details	19
Authors' Addresses	19

1. Requirements Language

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

2. Introduction

2.1. General

This memo defines a Kerberos-based Security Model (KSM) that supports centralized administration of user authentication and authorization for SNMPv3 [RFC3411]. Encryption of the payload is optional, as is Integrity protection for the whole SNMP message. Kerberos is defined in [RFC4120].

SNMPv3 allows for the definition of new security models. The first security models, USM [RFC3414] and VACM [RFC3415], do not support centralized security administration. Kerberos is a widely-deployed security infrastructure that provides centrally administered authentication, authorization and message protection for large scale networks. This model allows SNMP to leverage that infrastructure.

Kerberos is a "trusted third party" security system. It uses symmetric key cryptography and is simple enough to be implemented on small managed devices like modems and home routers.

This memo describes Kerberos related data structures and PDUs only at a level necessary for an explanation of the model. It is assumed that readers are already familiar with the SNMPv3 [RFC3411] and Kerberos [RFC4120] protocols. Specific details for implementing KSM are included in the Appendix A.

This memo addresses security for all SNMP operations and discusses how a SNMP command responder can use the user authorization information in RADIUS [RFC5608] servers to make authorization decisions for requested SNMP operations.

2.2. Use Cases

The following are some of the less common use cases that serve as a basis for generating requirements for this security model.

- o The managed devices may not be trustable because of their physical location in untrusted areas, like in customers' homes. Specifically, they cannot be trusted with information that can be used to access other managed devices, for example, the userids and

passwords of the administrators that manage these devices.s

- o The managed devices may be low end devices like modems and wireless routers in customers' homes. They may have limited CPU and memory capabilities.
- o Large numbers (millions) of managed devices may be polled for status periodically and frequently by automated (unmanned) status monitoring programs.

2.3. Security Model Requirements

The SNMP Architecture [RFC3411] lists the following well understood requirements:

- REQ1: Modification of Information: The modification threat is the danger that an unauthorized entity may alter in-transit SNMP messages generated on behalf of an authorized principal in such a way as to effect unauthorized management operations, including falsifying the value of an object.
- REQ2: Masquerade: The masquerade threat is the danger that management operations not authorized for some user may be attempted by assuming the identity of another user that has the appropriate authorizations.
- REQ3: Message Stream Modification: The SNMP protocol is typically based upon a connection-less transport service which may operate over any sub-network service. The re-ordering, delay or replay of messages can and does occur through the natural operation of many such sub-network services. The message stream modification threat is the danger that messages may be maliciously re-ordered, delayed or replayed to an extent which is greater than can occur through the natural operation of a sub-network service, in order to effect unauthorized management operations.
- REQ4: Disclosure: The disclosure threat is the danger of eavesdropping on the exchanges between managed agents and a management station. Protecting against this threat may be required as a matter of local policy.

The above requirements can be restated as the need for user authentication, message encryption, message integrity and message replay protection over UDP.

In addition, many environments often require additional operational constraints, such as:

- REQ5: Centralized security administration: User credentials for both authentication and authorization need to be administered from a central point.
- REQ6: Convenience: it must not be necessary to for an administrator to reauthenticate to each and every device that is accessed, unless subsequent access to a device requires stronger authentication.
- REQ7: Strong Authentication: Two factor authentication must be supported for access to sensitive devices. However, the administrator should not be required to reauthenticate for each and every device that is accessed.
- REQ8: Efficiency: The model should support low end managed devices that often have constraints such as, inability to save state between SNMP commands, and the lack of computational capability that is required to support PKI based protocols. Small devices optimally exchange messages using transports such as UDP. The set up and tear down overheads for connections between the SNMP command generators and command responders should be minimal. The connections may be short-lived and should not be required to be persistent.

2.4. Message Protection Requirements

The model should support the levels of security as described in the SNMP Architecture:

- o noAuthNoPriv - without authentication and without privacy. This option will not protect either the SNMP message or header.
- o authNoPriv - with authentication but without privacy. Both the SNMP request and response will be authenticated. The entire message (including the SNMP header) will be integrity protected.
- o authPriv - with both authentication and privacy. Both the SNMP request and response will be authenticated. The entire message (including the SNMP header) will be integrity protected and the scopedPDU will be further protected for confidentiality.

2.5. Kerberos Service Operational Model

Kerberos uses the concept of shared secrets to allow principals to securely communicate with each other. Principals may be human users (typically administrators) or hosts (both managing and managed devices.) Each principal has a unique name and shares it's unique secret key with the Kerberos server. For human users, the key is

derived from a password that is entered by the user during login. For hosts, the secret key has to be generated during installation and is securely stored on the host. Note that a SNMP Manager (a command generator) may be an unattended daemon (e.g. an automated status poller,) or an interactive program run by an administrator.

Before attempting any SNMP exchanges, the Manager should obtain a Ticket Granting Ticket (TGT) from the Kerberos Key Distribution Center (KDC). The descriptions below assume that this step has been completed. The SNMP command generators use the TGT to obtain Kerberos service tickets for each managed device that they subsequently accesses.

Note that the command generator has to get the TGT only once before accessing multiple command responders. There is a concept of "single signon" inherent in this model. If the installation's policy requires it, then two factor (token based) authentication for administrators should be performed at this point. Any interactivity required for resynchronizing tokens can be easily handled in this situation. Kerberos must set a flag in the TGT indicating that the required authentication was completed. This flag can be checked before an SNMP request is performed.

2.6. User Authorization Information

The Kerberos ticket can contain user authorization information. This information is inserted into the AuthorizationData field of the encrypted ticket by the Kerberos server and cannot be modified by others. It is an indication of the privileges assigned to the requesting user.

For KSM, this value must contain the permissions for the specific user on the identified device. Though the semantics of this value can vary widely, for SNMP it is constrained to be a VACM groupName that the user should be placed into. This value can also be a role, privilege, or permission bits, without change in the syntax.

This information will be configured and maintained centrally in an Authorization Database, and not on each individual managed device. Each managed device must be aware of the privileges associated with that groupName.

3. Elements of the model

3.1. How KSM fits into the SNMP Architecture

Figure 1 shows the positioning of the KSM Module in a SNMP command responder. KSM is shown as a peer to the User-based Security Model (USM). The positioning of the KSM Module in a SNMP command generator is similar.

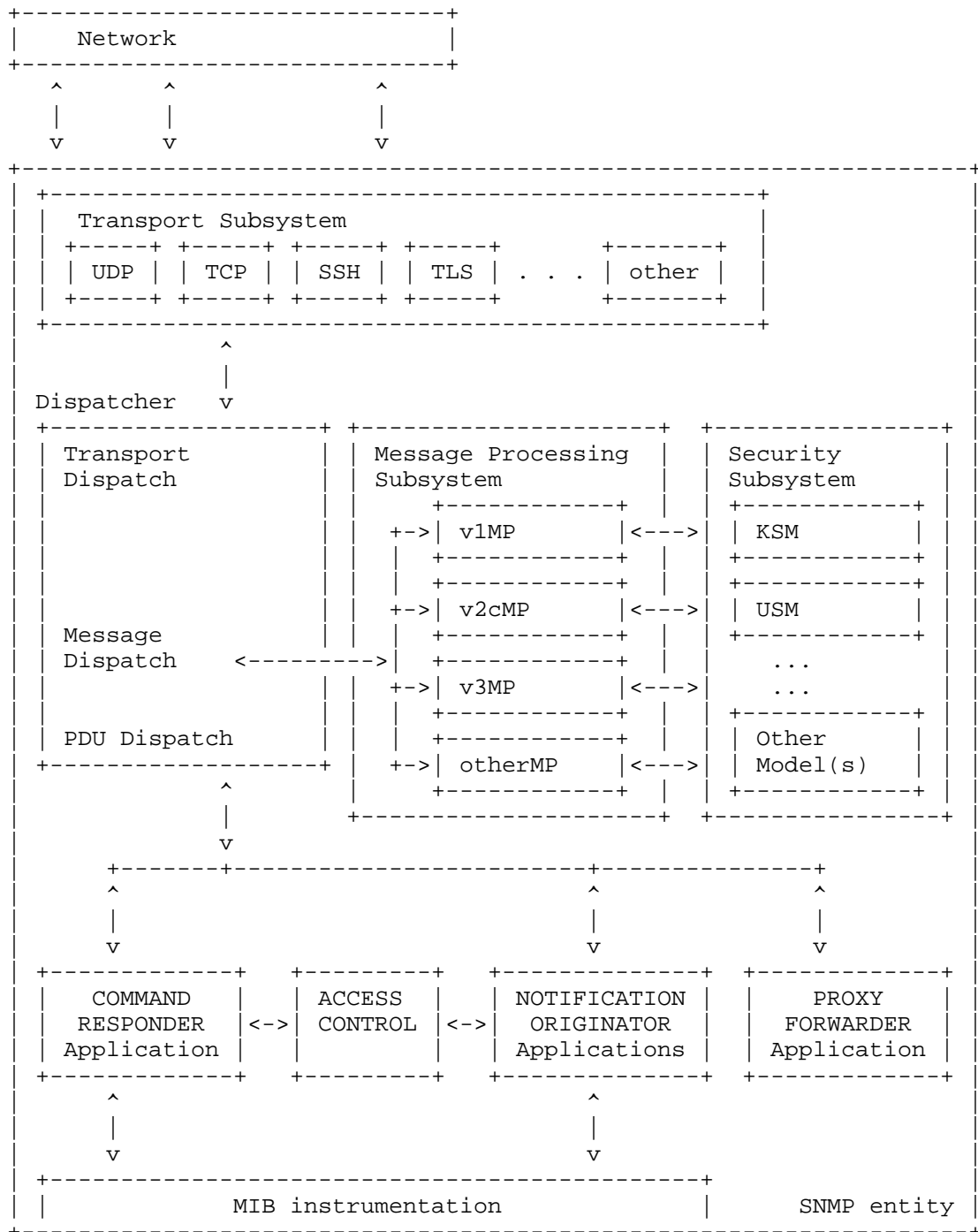


Figure 1: SNMP Command Responder Architecture

In Figure 1, a SNMPv3 message may reach the SNMP command responder using any supported transport. For example, UDP is an expected common transport to be used with KSM. The Message Dispatcher forwards the message to the Message Processing Subsystem. If the msgSecurityModel specifies KSM, then the message is forwarded to the KSM module.

3.2. High Level Architecture

Figure 2 depicts the relationship between the major components of the proposed solution. The dashed lines indicate message exchanges while dotted lines indicate a shared secret.

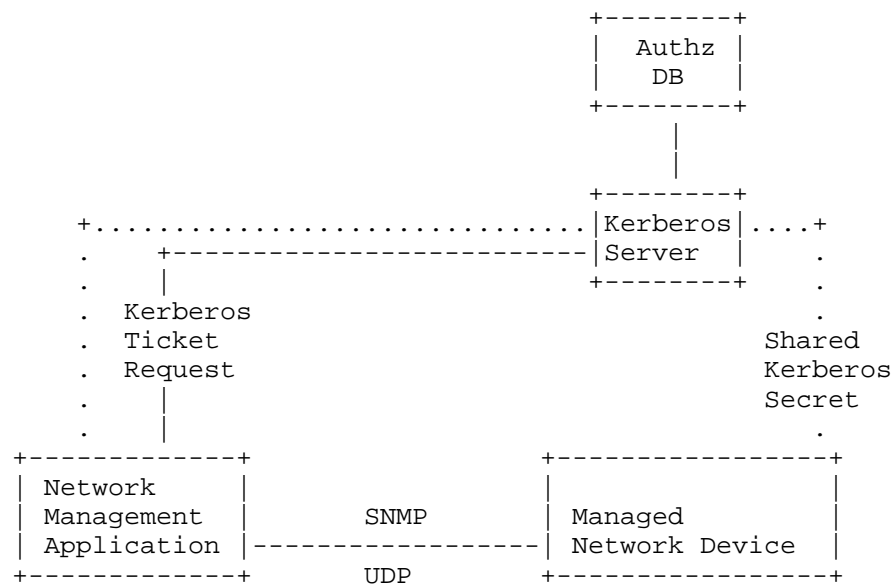


Figure 2: High-level Architecture

The Authorization Database is a centralized repository that contains a mapping of access privileges for each user to each device. It is accessed securely by the Kerberos server. Note that details of the Kerberos Server and the Authorization Database are out of scope of this document.

The following sequence summarizes most SNMP request and response exchanges. They are detailed in subsequent sections.

1. The SNMP command generator first obtains a service ticket for the specific command responder from the KDC. The Kerberos server may look up the authorization information for the requestor to the managed device by checking the authorization database. This information may be inserted into the service ticket, which is returned to the requestor.
2. The SNMP command generator sends a SNMPv3 request to the SNMP command responder. This request contains a Kerberos AP_REQ in the SNMPv3 securityParameters field.
3. When the SNMP command responder receives the SNMPv3 request, it must check the Session Key and the Lifetime of the ticket contained inside the AP_REQ. If the ticket has expired, the SNMP command responder must reject the SNMPv3 request.
4. The SNMP command responder responds with a SNMPv3 reply. This reply must contain a Kerberos AP_REP in the SNMPv3 securityParameters field.
5. The SNMP command generator receives the reply and must validate the Kerberos AP_REP.

Subsequent SNMP requests sent from the SNMP command generator to the same SNMP command responder can reuse the ticket obtained in step 1. A fresh AP_REQ will be constructed using the previously obtained information.

3.3. Outline of the Proposal

The msgSecurityModel field in the SNMPv3 packet header must contain a new value indicating the use of the Kerberos Security Model. The SNMPv3 securityLevel field must contain a value indicating whether or not authentication and privacy are being requested. The SNMPv3 securityParameters field must contain a Kerberos AP_REQ in a SNMP request and must contain a Kerberos AP_REP in a SNMP response. These Kerberos structures contain service tickets, which in turn contain an encryption key. The encryption key must be used to protect the scopedPDU. The Kerberos ticket may also contain user authorization information (like a GroupName) in the AuthorizationData field. This information may be used by VACM in the SNMP command responder to make an authorization decision.

3.4. Services for generating an outgoing SNMP request

When a SNMP command generator needs to make a SNMP request, it must generate an SNMP PDU. The Security Subsystem is invoked after the SNMP PDU is constructed. As described in [RFC3411], the Message

Processing subsystem invokes the Security Subsystem using the following interface provided by KSM

```
statusInformation =
  generateRequestMsg(
    IN  messageProcessingModel  -- typically, SNMP version
    IN  globalData              -- message header, admin data
    IN  maxMessageSize          -- of the sending SNMP entity
    IN  securityModel           -- for the outgoing message
    IN  securityEngineID        -- authoritative SNMP entity
    IN  securityName            -- on behalf of this principal
    IN  securityLevel           -- Level of Security requested
    IN  scopedPDU               -- message (plaintext) payload
    OUT securityParameters      -- filled in by Security Module
    OUT wholeMsg                -- complete generated message
    OUT wholeMsgLength          -- length of the generated message
  )
```

Figure 3

- o securityModel: Must be Kerberos Security Model (KSM).
- o securityName: The name of the principal requesting the command. This name will identify the Kerberos credentials (TGT) that will be used.
- o securityLevel: This determines if the message needs to be protected from disclosure and if the message needs to be authenticated.
- o scopedPDU: The payload will be protected as needed and will be copied as part of the wholeMsg.
- o securityParameters: This is filled in by KSM.

3.5. Services for generating an outgoing SNMP response

Similarly, when a SNMP command responder needs to respond to a SNMP request, it must generate an SNMP response PDU. KSM provides the following interface:

```

statusInformation =          -- success or errorIndication
generateResponseMsg(
  IN  messageProcessingModel  -- typically, SNMP version
  IN  globalData              -- message header, admin data
  IN  maxMessageSize          -- of the sending SNMP entity
  IN  securityModel           -- for the outgoing message
  IN  securityEngineID        -- authoritative SNMP entity
  IN  securityName            -- on behalf of this principal
  IN  securityLevel           -- Level of Security requested
  IN  scopedPDU               -- message (plaintext) payload
  IN  securityStateReference  -- reference to security state
                                -- information from original
                                -- request
  OUT securityParameters      -- filled in by Security Module
  OUT wholeMsg                -- complete generated message
  OUT wholeMsgLength          -- length of generated message
)

```

Figure 4

- o securityStateReference: A handle/reference to cachedSecurityData that was generated when processing the incoming Request message to which this is the Response message.

3.6. Services for processing an incoming SNMP message

When the SNMP command responder receives a SNMP request, the message is passed through the Transport Dispatcher and the Message Processing Subsystem to the KSM Module. As described in [RFC3411], the Message Processing Subsystem invokes the Security Subsystem using the following interface provided by KSM:


```

statusInformation =          -- errorIndication or success
                             -- error counter OID/value if error

    processIncomingMsg(
    IN  messageProcessingModel -- typically, SNMP version
    IN  maxMessageSize        -- of the sending SNMP entity
    IN  securityParameters    -- for the received message
    IN  securityModel         -- for the received message
    IN  securityLevel         -- Level of Security
    IN  wholeMsg              -- as received on the wire
    IN  wholeMsgLength        -- length as received on the wire
    OUT securityEngineID      -- authoritative SNMP entity
    OUT securityName          -- identification of the principal
    OUT scopedPDU,            -- message (plaintext) payload
    OUT maxSizeResponseScopedPDU -- maximum size sender can handle
    OUT securityStateReference -- reference to security state
    )                          -- information, needed for response

```

Figure 5

- o securityParameters: These are the security parameters as received in the message.
- o securityModel: Must be Kerberos Security Model.
- o securityName: The name of the principal requesting the command is filled in by KSM.
- o securityStateReference: A handle/reference to cachedSecurityData to be used when securing an outgoing Response message..

4. Elements of the procedure

4.1. Procedure for outgoing requests

This section describes the procedure followed by an SNMP engine whenever it generates a message containing an outgoing request (like a request, a notification, or a report) on behalf of a user, with a particular securityLevel.

If the securityLevel is noAuthNoPriv, then the scopedPDU is simply copied as part of the output wholeMsg.

The Kerberos credentials are located using the specified securityName. The Kerberos principal name of the command responder service is determined. This should be composed of the service name "snmp" and the hostname. The hostname should be the textual representation of the IP address (v4 or v6). For example,

"snmp/192.168.1.25@realm.com".

A fresh Kerberos AP_REQ must be constructed for each message transmission and retransmission. Note that Kerberos will reuse a suitable unexpired service ticket if it is already available. If SNMP authentication is required, then the Kerberos Mutual authentication option must be requested.

If the securityLevel of the message is set to AuthPriv, the scopedPdu payload must be encrypted using the encryption key in the AP_REQ message.

The ksmSecurityParameters is constructed as follows.

```
ksmSecurityParameters ::= SEQUENCE {  
  -- The Kerberos 5 checksum type used to checksum this message  
  ksmChecksumType      INTEGER(0..2147483647),  
  -- The actual keyed checksum data returned by Kerberos  
  ksmChecksum          OCTET STRING,  
  -- The Kerberos 5 message (AP_REQ)  
  ksmKerberosMsg       OCTET STRING,  
}
```

Figure 6

- o ksmChecksumType is an integer which corresponded to the checksum algorithm used to secure this message as defined by Kerberos.
- o ksmChecksum is the space for the checksum and is initialized with the value 0.
- o ksmKerberosMsg is the Kerberos 5 AP_REQ message.

The checksum algorithm defined by ksmChecksumtype is run and resulting value is placed into ksmChecksum. The ASN.1 structure is BER encoded as an OCTET STRING and copied into the SNMPv3 msgSecurityParameters.

4.2. Procedure for incoming requests

This section describes the procedure followed by an SNMP engine whenever it receives a message containing a management operation on behalf of a user, with a particular securityLevel.

The securityParameters value must be deserialized. The ksmKerberosMsg value must contain a Kerberos AP_REQ message. This message must be processed and the service ticket must be extracted.

The Kerberos principal name of the requestor must be obtained from the ticket. This name must be returned as the value of `securityName`.

If present, the requesting user's authorization information is also extracted from the ticket and associated with the `securityStateReference`.

If the `securityLevel` of the message is set to `AuthPriv`, the payload must be decrypted using the encryption key in the `AP_REQ` message.

Lastly, the integrity of the entire message must be checked.

4.3. Procedure for authorizing incoming requests

This section describes the procedures followed by an Access Control module that implements the View-based Access Control Model when checking access rights as requested by an application (for example a Command Responder or a Notification Originator application). The Access Control subsystem invokes the View-based Access Control Model with the following primitive:

```
statusInformation =          -- success or errorIndication
  isAccessAllowed(
    IN  securityModel          -- Security Model in use
    IN  securityName          -- principal who wants to access
    IN  securityLevel          -- Level of Security
    IN  viewType              -- read, write, or notify view
    IN  contextName           -- context containing variableName
    IN  variableName          -- OID for the managed object
  )
```

Figure 7

The `securityName`, `securityModel` and the authorization information that was extracted from the ticket are used to identify a `groupName` that will be used to make the access control decision. If the authorization information is the `groupName`, then it can be directly used.

4.4. Procedure for outgoing responses

After the SNMP command has been completed, the results need to be communicated back to the requestor. This procedure is similar to that for the outgoing request.

A fresh Kerberos `AP_REP` must be constructed for each message response.

If the securityLevel of the message is set to AuthPriv, the scopedPdu payload must be encrypted using the encryption key received earlier.

The ksmSecurityParameters is constructed as described earlier, except that the AP_REP message is used instead of the AP_REQ message. If authentication is requested, a message checksum is computed as described earlier.

4.5. Procedure for incoming responses

Lastly, this section describes the procedures for processing response messages on the command generator. The procedure is similar to that of processing incoming requests, except that a Kerberos AP_REP message appears instead of the AP_REQ.

5. Unconfirmed messages

Traps and Reports are handled just as all other outgoing requests. A fresh AP_REQ must be constructed for each message. No response is expected or processed.

InformsRequests are confirmed messages and are handled like other confirmed requests. A response is expected and processed.

6. Comparative Analysis

6.1. Addressing the requirements

KSM uses the standard Kerberos protocols to effectively address the following basic security requirements: Modification of Information, Masquerade, Message Stream Modification and Disclosure.

Typical Kerberos implementations do not support two factor token based authentication, but at least one such implementation has been built and deployed in a large scale.

6.2. Advantages of KSM

- o Small Number of UDP packet exchanges.
- o SNMP command responder can make an authorization decision without communicating with any other service.
- o SNMP command responder does not need to retain any state information between SNMP requests.

- o Compatible with VACM.

6.3. Disadvantages of this Security Model

Tickets are device specific, so the SNMP command generator must obtain a ticket for each command responder before SNMP exchanges are started. Hence, the generator must get a larger number of tickets if it wants to communicate with a large number of SNMP command generators devices. However, this burden is placed on the command generator and not on the managed device.

A fresh AP_REQ (AP_REP) must be constructed for each command (response) generated. However, only the first AP_REQ will require communications with the Kerberos server to get a service ticket.

6.4. Issues with models based on transport sessions

One of the approaches to SNMP security is to set up a secure "transport" connection using SSH between the SNMP command generator and the SNMP command responder. Some of the issues with this approach are:

1. Every SNMP command responder needs to be trusted since it sees administrator passwords. It would be tempting for a hacker to spoof a SNMP command responder and steal the passwords, which could then be used to attack other managed devices.
2. Every SNMP command responder needs to be configured as a RADIUS client.
3. User authentication hits the RADIUS server for each device accessed. Ideally, RADIUS should be invoked only once during user "login".
4. Token based two-factor authentication is not fully supported because interactivity may be required for resynchronization and 'new PIN' modes.
5. Setup overhead for ssh (and TLS) precludes short SNMP exchanges (like status polls.)
6. SNMP command responder code is more complicated. It has to retain much more state.
7. SSH (or TLS) uses public key cryptography during session setup, which is somewhat compute intensive.

7. IANA Considerations

The following IDs and constants should be allocated:

1. SNMPv3 SecurityModel ID for Kerberos Security Model (KSM)
2. Kerberos AuthorizationData type ID indicating that the content is SNMP Authorization information.
3. Kerberos PreAuthentication data type indicating that the content is requesting that SNMP Authorization information be added.

8. Security Considerations

This model depends on Kerberos and inherits its security attributes. Major considerations are:

1. The secret keys stored on Managed Devices are often hard to protect and manage. This is especially true when the devices are physically located in untrusted zones. However, an attacker that copies the secret key is limited to impersonating the device and cannot affect other devices in the network.
2. The model depends on the Kerberos server being available online to generate service tickets. If the server becomes unavailable then SNMP operations will be affected. However, the Kerberos server can be replicated, and service tickets that have already been issued will continue to work until they expire. SNMP command generators may wish to request service tickets ahead of needing them to ensure that access to a device is possible even if network outages occur between the command generators and the KDC.

9. Acknowledgements

Authors would like to acknowledge the previous internet draft [draft-hornstein-snmpv3-ksm].

10. References

10.1. Normative References

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

- [RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, RFC 3411, December 2002.
- [RFC3414] Blumenthal, U. and B. Wijnen, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)", STD 62, RFC 3414, December 2002.
- [RFC3415] Wijnen, B., Presuhn, R., and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)", STD 62, RFC 3415, December 2002.
- [RFC4120] Neuman, C., Yu, T., Hartman, S., and K. Raeburn, "The Kerberos Network Authentication Service (V5)", RFC 4120, July 2005.
- [RFC5608] Narayan, K. and D. Nelson, "Remote Authentication Dial-In User Service (RADIUS) Usage for Simple Network Management Protocol (SNMP) Transport Models", RFC 5608, August 2009.

10.2. Informative References

- [I-D.ietf-isms-radius-vacm]
Narayan, K., Nelson, D., and R. Presuhn, "Using Authentication, Authorization, and Accounting services to Dynamically Provision View-based Access Control Model User-to-Group Mappings", draft-ietf-isms-radius-vacm-11 (work in progress), September 2010.

Appendix A. Implementation Details

Sample code for KSM implementation: TBD.

Authors' Addresses

Rajaram Pejaver
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: rajaram_pejaver@cable.comcast.com
URI: <http://www.comcast.com>

Yiu L. Lee
Comcast
One Comcast Center
Philadelphia, PA 19103
U.S.A.

Email: yiul_lee@cable.comcast.com
URI: <http://www.comcast.com>

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

Phone: +1 530 792 1913
Email: ietf@hardakers.net
URI: <http://www.hardakers.net/>

Ken Hornstein
US Naval Research Laboratory
Bldg A-49, Room 2, 4555 Overlook Avenue
Washington DC 20375
U.S.A.

Phone: +1 (202) 404-4765
Email: Kenh@cmf.nrl.navy.mil

