

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.

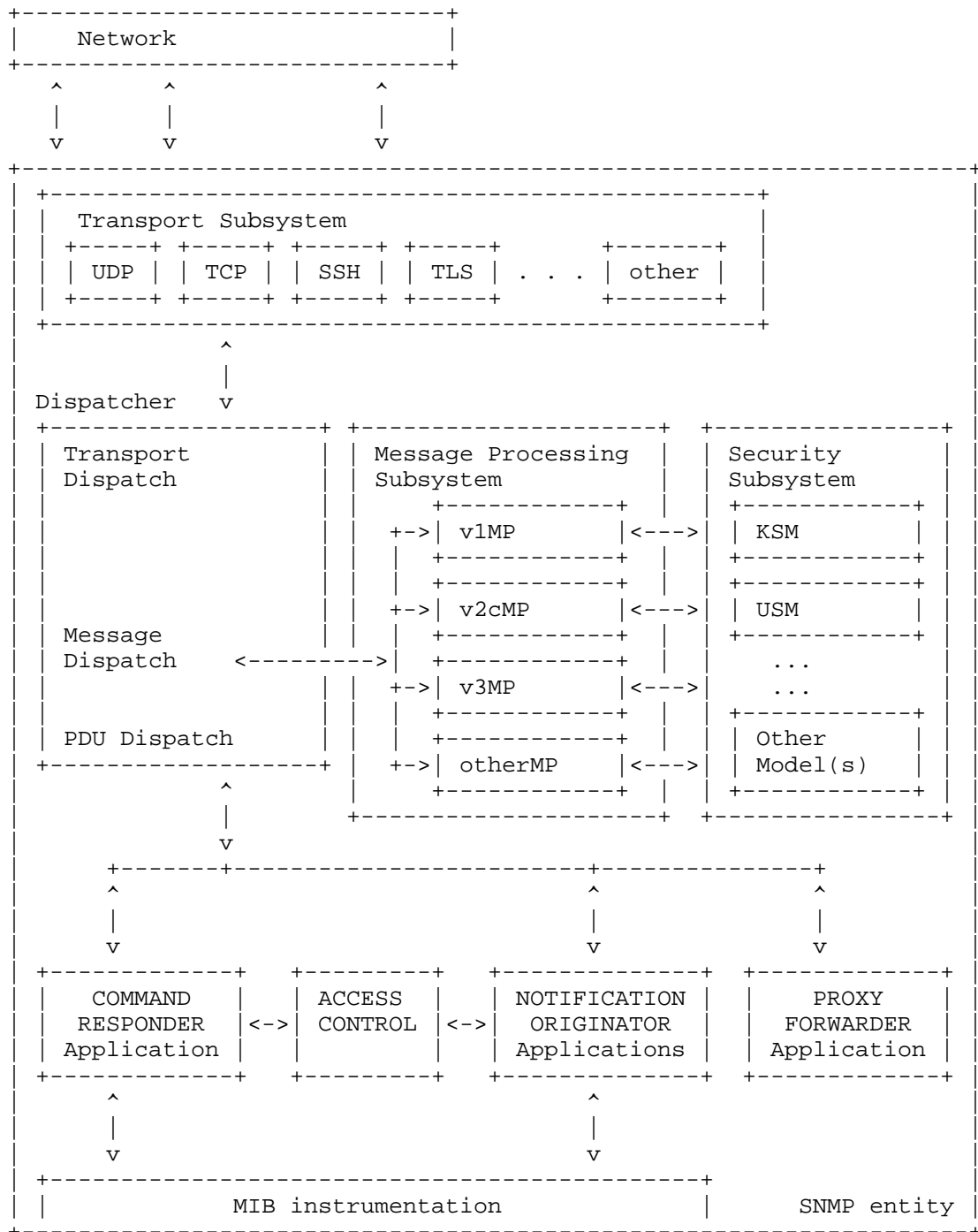




Figure 1: SNMP Command Responder Architecture

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.

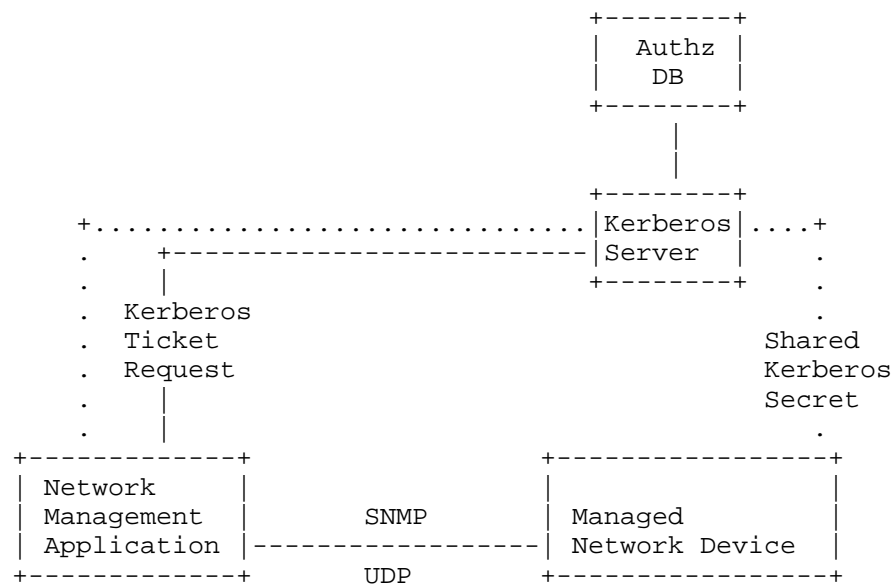


Figure 2: High-level Architecture

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.

InformRequests 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](mailto: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](mailto: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](mailto: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](mailto:Kenh@cmf.nrl.navy.mil)

