Groupe de travail Réseau
Request for Comments : 3415
STD : 62
RFC rendue obsolète : 2575
Catégorie : Norme

B. Wijnen, Lucent Technologies
R. Presuhn, BMC Software, Inc.
K. McCloghrie, Cisco Systems, Inc.
décembre 2002
Traduction Claude Brière de L'Isle

Modèle de contrôle d'accès fondé sur la vue (VACM) pour le protocole simple de gestion de réseau (SNMP)

Statut du présent mémoire
Le présent document spécifie un protocole de l'Internet en cours de normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé
Le présent document décrit le modèle de contrôle d'accès fondé sur la vue (VACM, *View-based Access Control Model)* à utiliser dans l'architecture du protocole simple de gestion de réseau (SNMP, *Simple Network Management Protocol*). Il définit les éléments de procédure pour contrôler l'accès aux informations de gestion. Le présent document inclut aussi une base de données d'informations de gestion (MIB, *Management Information Base*) pour la gestion à distance des paramètres de configuration pour le modèle de contrôle d'accès fondé sur la vue. Le présent document rend obsolète la RFC2575.

Table des Matières

1. Introduction

L'architecture pour la description des cadres de gestion de l'Internet [RFC3411] indique qu'un moteur SNMP est composé :
1) d'un expéditeur,
2) d'un sous-système de traitement de message,
3) d'un sous-système de sécurité, et
4) d'un sous-système de contrôle d'accès.

Les applications utilisent les services de ces sous-systèmes.

Il est important de comprendre l'architecture SNMP et sa terminologie pour comprendre comment le modèle de contrôle d'accès fondé sur la vue décrit dans le présent document s'intègre dans l'architecture et interagit avec les autres sous-systèmes au sein de l'architecture. Le lecteur est supposé avoir lu et compris la description et la terminologie de l'architecture SNMP, telle que définie dans la [RFC3411].

Le sous-système de contrôle d'accès d'un moteur SNMP est chargé de vérifier si un type spécifique d'accès (lecture, écriture, notification, est permis sur un objet particulier (une instance).

L'objet du présent document est de définir un modèle spécifique du sous-système de contrôle d'accès, appelé "modèle de contrôle d'accès fondé sur la vue". Noter qu'il ne s'agit pas nécessairement du seul modèle de contrôle d'accès.

Dans le présent document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP 14, [RFC2119].

1.2 Contrôle d'accès

Le contrôle d'accès se fait (implicitement ou explicitement) dans une entité SNMP lors du traitement des messages SNMP de demande de restitution ou de modification de la part d'une entité SNMP. Par exemple, une application de répondeur de commandes applique le contrôle d'accès lorsque elle traite les demandes qu'elle reçoit d'une application de générateur de commandes. Ces demandes contiennent des PDU de classe Lecture et Écriture, comme défini dans la [RFC3411].

Le contrôle d'accès se fait aussi dans une entité SNMP lorsque un message de notification SNMP est généré (par une application de générateur de notifications). Ces messages de notification contiennent des PDU de classe Notification comme défini dans la [RFC3411].

Le modèle de contrôle d'accès fondé sur la vue définit un ensemble de services qu'une application (comme une application de répondeur de commandes ou de générateur de notifications) peut utiliser pour vérifier les droits d'accès. Il est de la responsabilité de l'application de faire les invocations de service appropriées pour les vérifications d'accès.

1.3 Magasin de configuration locale

Pour mettre en œuvre le modèle décrit dans le présent document, une entité SNMP doit conserver les informations sur les droits d'accès et les politiques. Ces informations font partie du magasin local de données de configuration (LCD, *Local Configuration Datastore*) du moteur SNMP. Voir dans la [RFC3411] la définition du LCD.

Pour permettre de configurer à distance le LCD d'une entité SNMP, des portions du LCD doivent être accessibles comme objets gérés. Un module de MIB, la MIB de configuration du modèle de contrôle d'accès fondé sur la vue, qui définit ces types d'objets gérés est inclus dans le présent document.

2. Éléments du modèle

Cette section contient les définitions pour réaliser le service de contrôle d'accès fourni par le modèle de contrôle d'accès fondé sur la vue.

2.1 Groupes

Un groupe est un ensemble de zéro, une ou plusieurs paires <securityModel, securityName> au nom desquelles on peut accéder aux objets de gestion SNMP. Un groupe définit les droits d'accès offerts à tous les noms de sécurité qui appartiennent à ce groupe. La combinaison d'un modèle de sécurité et d'un nom de sécurité se transpose au plus en un groupe. Un groupe est identifié par un groupName *(nom de groupe)*.

Le module de contrôle d'accès suppose que le securityName *(nom de sécurité)* a déjà été authentifié comme nécessaire et ne fournit par lui-même pas d'autre authentification.

Le modèle de contrôle d'accès fondé sur la vue utilise le securityModel *(modèle de sécurité)* et le securityName comme entrées au module de contrôle d'accès lorsque il est invoqué pour vérifier les droits d'accès. Il détermine le groupName comme une fonction du modèle de sécurité et du nom de sécurité.

2.2 Niveau de sécurité

Différents droits d'accès peuvent être définis pour les membres d'un groupe pour différents niveaux de sécurité, c'est-à-dire, noAuthNoPriv, authNoPriv, et authPriv. Le securityLevel *(niveau de sécurité)* identifie le niveau de sécurité qui va être supposé lors de la vérification des droits d'accès. Voir dans le document d'architecture SNMP [RFC3411] la définition de securityLevel.

Le modèle de contrôle d'accès fondé sur la vue exige que le securityLevel soit passé en entrée au module de contrôle d'accès lorsque il est invoqué pour vérifier les droits d'accès.

2.3 Contextes

Un contexte SNMP est une collection d'informations de gestion accessibles par une entité SNMP. Un élément d'informations de gestion peut exister dans plus d'un contexte. Une entité SNMP a potentiellement accès à de nombreux contextes. On trouvera les détails de la dénomination des informations de gestion dans le document d'architecture SNMP [RFC3411].

Le modèle de contrôle d'accès fondé sur la vue définit un tableau "vacmContextTable" qui fait la liste des contextes disponibles localement par contextName *(nom de contexte)*.

2.4 Vues de MIB et familles de vues

Pour des raisons de sécurité, il est souvent précieux d'être capable de restreindre les droits d'accès de certains groupes à un sous ensemble des informations de gestion dans le domaine de gestion. Pour fournir cette capacité, l'accès à un contexte se fait via une "vue de MIB" qui détaille un ensemble spécifique de types d'objets gérés (et facultativement, les instances spécifiques de types d'objets) au sein de ce contexte. Par exemple, pour un certain contexte, il va normalement toujours y avoir une vue de MIB qui fournit l'accès à toutes les informations de gestion dans ce contexte, et souvent il y aura d'autres vues de MIB dont chacune contient un sous ensemble des informations. Ainsi, l'accès permis à un groupe peut être restreint de la manière désirée en spécifiant ses droits sous la forme de la vue de MIB particulière (sous ensemble) à laquelle il peut accéder au sein de chaque contexte approprié.

Comme les types d'objets gérés (et leurs instances) sont identifiés via la structure de désignation en forme d'arborescence d'IDENTIFIANT D'OBJET de l'ISO [ISO-ASN.1], [RFC2578], il est pratique de définir une vue de MIB comme la combinaison d'un ensemble de "sous arborescences de vues", où chaque sous arborescence de vue est une sous arborescence au sein de l'arborescence de désignation de l'objet géré. Donc, une simple vue de MIB (par exemple, tous les objets gérés au sein du cadre de gestion de réseau de l'Internet) peuvent être définis comme une seule arborescence de vue, tandis que des vues de MIB plus compliquées (par exemple, toutes les informations pertinentes pour une certaine interface réseau) peuvent être représentées par l'union de plusieurs sous arborescences de vue.

Bien que tout ensemble d'objets gérés puisse être décrit par l'union d'un certain nombre de sous arborescences de vues, il peut se produire des situations qui vont exiger un très grand nombre de sous arborescences de vues. Cela pourrait arriver, par exemple, lorsque on spécifie toutes les colonnes d'une rangée conceptuelle d'un tableau de MIB parce que elles vont apparaître dans des sous arborescences séparées, une par colonne, chacune d'un format très similaire. Parce que les formats sont similaires, l'ensemble de sous arborescences requis peut aisément être agrégé en une seule structure. Cette structure est appelée une famille de sous arborescences de vues d'après l'ensemble de sous arborescences qui est représenté conceptuellement. Une famille de sous arborescences de vues peut être incluse ou exclue d'une vue de MIB.

2.4.1 Sous arborescences de vues

Une sous arborescence de vues est l'ensemble de toutes les instances d'objet de MIB qui ont un préfixe d'IDENTIFIANT D'OBJET ASN.1 commun à leur nom. Une sous arborescence de vues est identifiée par la valeur d'IDENTIFIANT D'OBJET qui est le plus long préfixe d'IDENTIFIANT D'OBJET commun à toutes les instances d'objet de MIB (potentielles) dans cette sous arborescence.

2.4.2 ViewTreeFamily

Une famille de sous arborescences de vues est un appariement d'une valeur d'IDENTIFIANT D'OBJET (appelée le nom de famille) avec une valeur de chaîne binaire (appelée le gabarit de famille). Le gabarit de famille indique quels sous identifiants du nom de famille associé sont significatifs pour la définition de la famille.

Pour chaque instance possible d'objet géré, cette instance appartient à une ViewTreeFamily particulière si les deux conditions suivantes sont vraies :

- le nom d'IDENTIFIANT D'OBJET de l'instance d'objet géré contient au moins autant de sous identifiants que le nom de famille, et
- chaque sous identifiant dans le nom d'IDENTIFIANT D'OBJET de l'instance d'objet géré correspond au sous identifiant correspondant du nom de famille chaque fois que le bit correspondant du gabarit de famille associé est différent de zéro.

Lorsque la valeur configurée du gabarit de famille est toute à un, la famille de sous arborescence de vue est identique à la seule sous arborescence de vue identifiée par le nom de famille.

Lorsque la valeur configurée du gabarit de famille est plus courte que ce qui est exigé pour effectuer la vérification ci-dessus, sa valeur est implicitement étendue avec des uns. Par conséquent, une famille de sous arborescence de vue qui a un gabarit de famille de longueur zéro correspond toujours à une seule sous arborescence de vue.

2.5 Politique d'accès

Le modèle de contrôle d'accès fondé sur la vue détermine les droits d'accès d'un groupe, représentant zéro, un ou plusieurs securityNames qui ont les mêmes droits d'accès. Pour un contexte particulier, identifié par contextName, auquel un groupe, identifié par groupName, a l'accès en utilisant un securityModel et un securityLevel particuliers, les droits d'accès de ce groupe sont donnés par une vue en lecture, une vue en écriture et une vue de notification.

La vue de lecture (read-view) représente l'ensemble des instances d'objets autorisés pour le groupe en lecture des objets. La lecture des objets survient lors du traitement d'une opération de restitution (lors du traitement des PDU de classe Read).

La vue d'écriture (write-view) représente l'ensemble des instances d'objet autorisé pour le groupe en écriture des objets. L'écriture des objets survient lors du traitement d'une opération d'écriture (lors du traitement des PDU de classe Write).

La vue de notification représente l'ensemble des instances d'objet autorisé pour le groupe lors de l'envoi d'objets dans une notification, comme lors de l'envoi d'une notification (lors de l'envoi de PDU de classe Notification).

3. Éléments de procédure

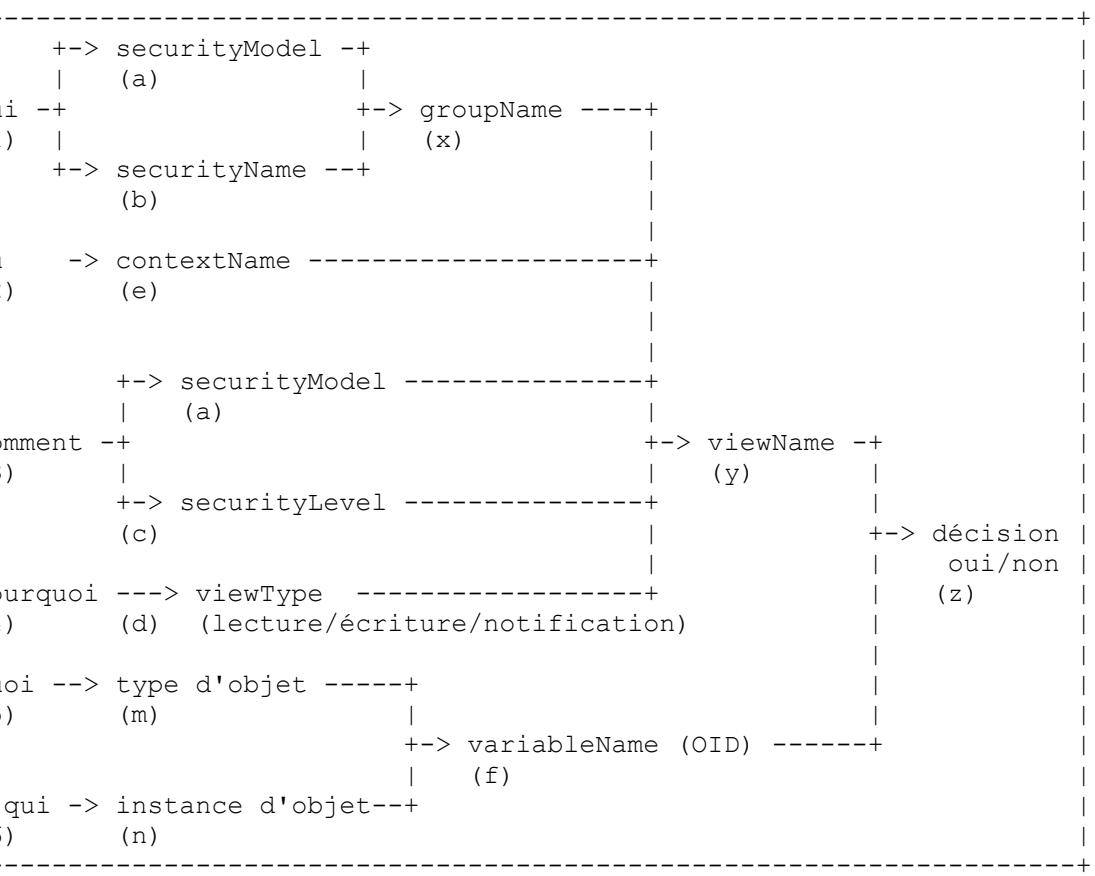
Cette section décrit les procédures suivies par un module de contrôle d'accès qui met en œuvre le modèle de contrôle d'accès fondé sur la vue lors de la vérification des droits d'accès demandée par une application (par exemple une application de répondeur de commandes ou une application de générateur de notifications). La primitive de service abstrait est :

```
statusInformation =             -- succès ou indication d'erreur
isAccessAllowed(
  securityModel                 -- modèle de sécurité utilisé
  securityName                  -- principal qui veut l'accès
  securityLevel                 -- niveau de sécurité
  viewType                      -- vue en lecture, écriture, ou notification
  contextName                   -- contexte qui contient le nom de variable
  variableName                  -- OID pour l'objet géré
  )
```

Les éléments de données abstraits sont :
statusInformation *(informations d'état)* : une des suivantes :
accessAllowed *(accès admis)* : une vue de MIB a été trouvée et l'accès est accordé.
notInView *(pas dans la vue)* : une vue de MIB a été trouvée mais l'accès est refusé. Le nom de la variable n'est pas dans la vue de MIB configurée pour le type de vue spécifié (par exemple, dans l'entrée pertinente du vacmAccessTable *(tableau d'accès VACM)*).
noSuchView *(pas de vue)* : aucune vue de MIB n'a été trouvée parce qu'aucune n'a été configurée pour le type de vue spécifié (par exemple, dans l'entrée pertinente du vacmAccessTable).
noSuchContext *(pas ce contexte)* : aucune vue de MIB n'a été trouvée parce que il n'y a pas d'entrée dans le vacmContextTable pour le nom de contexte spécifié.
noGroupName *(pas de nom de groupe)* : aucune vue de MIB n'a été trouvée parce que aucune entrée n'a été configurée dans le vacmSecurityToGroupTable pour la combinaison spécifiée de modèle de sécurité et de nom de sécurité.
noAccessEntry *(pas d'entrée d'accès)* : aucune vue de MIB n'a été trouvée parce que aucune entrée n'a été configurée dans le vacmAccessTable pour la combinaison spécifiée de nom de contexte, nom de groupe (d'après le vacmSecurityToGroupTable) modèle de sécurité et niveau de sécurité.
otherError *(autre erreur)* : échec, une erreur non définie s'est produite.
securityModel *(modèle de sécurité)* : modèle de sécurité sous lequel l'accès est demandé.
securityName *(nom de sécurité)* : le principal au nom duquel l'accès est demandé.
securityLevel *(niveau de sécurité)* : niveau de sécurité sous lequel l'accès est demandé.
viewType *(type de vue)* : vue à vérifier (lecture, écriture ou notification).
contextName *(nom de contexte)* : contexte dans lequel l'accès est demandé.
variableName *(nom de variable)* : instance d'objet auquel l'accès est demandé.

3.1 Vue d'ensemble du traitement de isAccessAllowed

Le schéma suivant montre comment la décision de contrôle d'accès est prise par le modèle de contrôle d'accès fondé sur la vue.



Comment est prise la décision pour isAccessAllowed *(l'accès est-il accordé)* :

1) Les entrées au service isAccessAllowed sont :
   (a) securityModel -- modèle de sécurité utilisé
   (b) securityName -- principal qui veut l'accès
   (c) securityLevel -- niveau de sécurité
   (d) viewType -- vue de lecture, écriture, ou notification
   (e) contextName -- contexte qui contient le nom de la variable
   (f) variableName -- OID pour l'objet géré ; ceci est constitué :
   - du type d'objet (m)
   - de l'instance d'objet (n)

2) Les "qui" partiels (1), représentés par le securityModel (a) et le securityName (b), sont utilisés comme indices (a, b) dans le tableau vacmSecurityToGroupTable pour trouver une seule entrée qui produit un groupe, représenté par groupName (x).

3) Le "où" (2), représenté par le contextName (e), le "qui", représenté par le groupName (x) de l'étape précédente, et le "comment" (3), représenté par securityModel (a) et securityLevel (c), sont utilisés comme indices (e, x, a, c) dans le tableau vacmAccessTable pour trouver une seule entrée qui contient trois vues de MIB.

4) Le "pourquoi" (4), représenté par le viewType (d), est utilisé pour choisir la vue de MIB appropriée, représentée par un viewName (y), tiré du tableau vacmAccessEntry choisi à l'étape précédente. Ce viewName (y) est un indice dans le tableau vacmViewTreeFamilyTable et choisit l'ensemble d'entrées qui définit les variableNames qui sont inclus dans ou exclus de la vue de MIB identifiée par le viewName (y).

5) La présence ou l'absence du type de données de gestion "quoi" (5) et de l'instance particulière "à qui" (6), représentés par le variableName (f), dans la vue de MIB sont alors vérifiées, par exemple, la décision oui/non (z).

3.2 Traitement de la demande de service isAccessAllowed

Ce paragraphe décrit la procédure suivie par un module de contrôle d'accès qui met en œuvre le modèle de contrôle d'accès fondé sur la vue chaque fois qu'il reçoit une demande isAccessAllowed.

1) Le tableau vacmContextTable est consulté pour avoir des informations sur le contexte SNMP identifié par le contextName. Si les informations sur ce contexte SNMP sont absentes du tableau, une errorIndication (noSuchContext) est alors retournée au module appelant.

2) Le tableau vacmSecurityToGroupTable est consulté pour transposer le modèle de sécurité et le nom de sécurité en nom de groupe. Si les informations sur cette combinaison sont absentes du tableau, une errorIndication (noGroupName) est alors retournée au module appelant.

3) Le tableau vacmAccessTable est consulté pour avoir des informations sur le groupName, contextName, securityModel et securityLevel. Si les informations sur cette combinaison sont absentes du tableau, une errorIndication (noAccessEntry) est alors retournée au module appelant.

4) a) si le viewType est "read", la vue de lecture est alors utilisée pour vérifier les droits d'accès ;
   b) si le viewType est "write", la vue d'écriture est alors utilisée pour vérifier les droits d'accès ;
   c) si le viewType est "notify", la vue de notification est alors utilisée pour vérifier les droits d'accès.
   Si la vue à utiliser est la vue vide (viewName de longueur zéro) une errorIndication (noSuchView) est alors retournée au module appelant.

5) a) Si aucune vue n'est configurée pour le viewType spécifié, une errorIndication (noSuchView) est alors retournée au module appelant.
   b) Si le variableName (instance d'objet) spécifié n'est pas dans la vue de MIB (voir la clause DESCRIPTION pour vacmViewTreeFamilyTable à la Section 4) une errorIndication (notInView) est alors retournée au module appelant.
   Autrement,
   c) Le variableName spécifié est dans la vue de MIB. Une statusInformation de succès (accessAllowed) est retournée au module appelant.

4. Définitions

DÉFINITIONS DE SNMP-VIEW-BASED-ACM-MIB ::= DÉBUT

IMPORTATIONS :
CONFORMITÉ DE MODULE, GROUPE D'OBJET DE SNMPv2-CONF
IDENTITÉ DE MODULE, TYPE D'OBJET, snmpModules DE SNMPv2-SMI
TestAndIncr, RowStatus, StorageType DE SNMPv2-TC
SnmpAdminString, SnmpSecurityLevel, SnmpSecurityModel DE SNMP-FRAMEWORK-MIB;

IDENTITÉ DE MODULE snmpVacmMIB
DERNIÈRE MISE À JOUR : "200210160000Z" -- 16 octobre 2002, minuit
ORGANISATION "groupe de travail SNMPv3"
INFORMATIONS DE CONTACT "messagerie du groupe de travail : snmpv3@lists.tislabs.com
Pour s'abonner : majordomo@lists.tislabs.com ; mettre dans le corps du message : subscribe snmpv3

Coprésident : Russ Mundy
Adresse postale : Network Associates Laboratories, 15204 Omega Drive, Suite 300, Rockville, MD 20850-4601, USA
mél : mundy@tislabs.com
téléphone : +1 301-947-7107

Coprésident : David Harrington
Adresse postale : Enterasys Networks, 35 Industrial Way, P. O. Box 5004, Rochester, New Hampshire 03866-5005, USA
mél : dbh@enterasys.com
téléphone : +1 603-337-2614

Co-éditeur : Bert Wijnen
Adresse postale : Lucent Technologies, Schagen 33, 3461 GL Linschoten, Netherlands
mél : bwijnen@lucent.com
téléphone : +31-348-480-685

Co-éditeur: Randy Presuhn
Adresse postale : BMC Software, Inc., 2141 North First Street, San Jose, CA 95131, USA
mél : randy_presuhn@bmc.com
téléphone : +1 408-546-1006

Co-éditeur : Keith McCloghrie
Adresse postale : Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706, USA
mél : kzm@cisco.com.
téléphone : +1-408-526-5260 "

DESCRIPTION "Définitions des informations de gestion pour le modèle de contrôle d'accès fondé sur la vue pour SNMP.
Copyright (C) The Internet Society (2002). Cette version de ce module de MIB fait partie de la RFC3415 ; voir dans la RFC elle-même les notices légales complètes."

-- **Historique des révisions**

REVISION : "200210160000Z" -- 16 octobre 2002, minuit
DESCRIPTION : "Précisions, publiée comme RFC3415"
REVISION : "199901200000Z" -- 20 janvier 1999, minuit
DESCRIPTION : "Précisions, publiée comme RFC2575"
REVISION : "199711200000Z" -- 20 novembre 1997, minuit
DESCRIPTION : "Version initiale, publiée comme RFC2275"
::= { snmpModules 16 }

-- **Allocations administratives** **********

vacmMIBObjects IDENTIFIANT D'OBJET ::= { snmpVacmMIB 1 }
vacmMIBConformance IDENTIFIANT D'OBJET ::= { snmpVacmMIB 2 }

-- Informations sur les contextes locaux **********

TYPE D'OBJET vacmContextTable
SYNTAXE : SEQUENCE DE VacmContextEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Tableau des contextes disponibles localement. Ce tableau fournit des informations aux applications SNMP de générateur de commandes afin qu'elles configurent de façon appropriée le vacmAccessTable pour contrôler l'accès à tous les contextes à l'entité SNMP. Ce tableau peut changer de façon dynamique si l'entité SNMP permet que des contextes soient ajoutés/supprimés de façon dynamique (par exemple lorsque sa configuration change). De tels changements ne vont survenir que si l'instrumentation de gestion à l'entité SNMP reconnaît plus (ou moins) de contextes. La présence d'entrées dans ce tableau et d'entrées dans le vacmAccessTable est indépendante. C'est-à-dire que un contexte identifié par une entrée dans ce tableau n'est pas nécessairement référencé par une entrée dans le vacmAccessTable ; et le ou les contextes référencés par une entrée dans le vacmAccessTable n'existent pas nécessairement actuellement et donc n'ont pas besoin d'être identifiés par une entrée dans ce tableau. Ce tableau doit être rendu accessible via le contexte par défaut afin que les applications de répondeur de commandes aient un moyen standard de restituer les informations. Ce tableau est en lecture seule. Il ne peut pas être configuré via SNMP."
::= { vacmMIBObjects 1 }

TYPE D'OBJET vacmContextEntry
SYNTAXE : VacmContextEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Informations sur un contexte particulier."
INDICE : { vacmContextName }
::= { vacmContextTable 1 }

VacmContextEntry ::= SEQUENCE { vacmContextName SnmpAdminString }

TYPE D'OBJET vacmContextName
SYNTAXE : SnmpAdminString (TAILLE(0 à 32))
MAX-ACCESS : lecture seule
STATUT : actuel
DESCRIPTION : "Nom lisible par l'homme qui identifie un certain contexte à une entité SNMP particulière. Le contextName vide (longueur zéro) représente le contexte par défaut."
::= { vacmContextEntry 1 }

-- Informations sur les groupes **********

TYPE D'OBJET vacmSecurityToGroupTable
SYNTAXE : SEQUENCE DE VacmSecurityToGroupEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Ce tableau transpose une combinaison de securityModel et securityName en un groupName qui est utilisé pour définir une politique de contrôle d'accès pour un groupe de principaux."
::= { vacmMIBObjects 2 }

TYPE D'OBJET vacmSecurityToGroupEntry
SYNTAXE : VacmSecurityToGroupEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Une entrée dans ce tableau transpose la combinaison d'un securityModel et d'un securityName en un groupName."
INDICE : { vacmSecurityModel, vacmSecurityName }
::= { vacmSecurityToGroupTable 1 }

VacmSecurityToGroupEntry ::= SEQUENCE
{
vacmSecurityModel SnmpSecurityModel,
vacmSecurityName SnmpAdminString,
vacmGroupName SnmpAdminString,

```
    vacmSecurityToGroupStorageType    StorageType,
    vacmSecurityToGroupStatus         RowStatus
  }
```

TYPE D'OBJET vacmSecurityModel
SYNTAXE : SnmpSecurityModel(1..2147483647)
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Modèle de sécurité par lequel est fourni le vacmSecurityName référencé par cette entrée. Noter que cet objet ne doit pas prendre la valeur 'any' (0)."
::= { vacmSecurityToGroupEntry 1 }

TYPE D'OBJET vacmSecurityName
SYNTAXE : SnmpAdminString (TAILLE(1..32))
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Le securityName pour le principal, représenté dans un format indépendant du modèle de sécurité, qui est transposé par cette entrée en un groupName."
::= { vacmSecurityToGroupEntry 2 }

TYPE D'OBJET vacmGroupName
SYNTAXE : SnmpAdminString (TAILLE(1..32))
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Nom du groupe auquel appartient cette entrée (par exemple, la combinaison de securityModel et securityName). Ce groupName est utilisé comme indice dans le vacmAccessTable pour choisir une politique de contrôle d'accès. Cependant, une valeur dans ce tableau n'implique pas qu'une instance ayant cette valeur existe dans le tableau vacmAccesTable."
::= { vacmSecurityToGroupEntry 3 }

TYPE D'OBJET vacmSecurityToGroupStorageType
SYNTAXE : StorageType
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Type de mémorisation pour cette rangée conceptuelle. Les rangées conceptuelles qui ont la valeur 'permanent' n'ont pas besoin de permettre l'accès en écriture à des objets des colonnes de la rangée."
DEFVAL : { nonVolatile }
::= { vacmSecurityToGroupEntry 4 }

TYPE D'OBJET vacmSecurityToGroupStatus
SYNTAXE : RowStatus
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Statut de cette rangée conceptuelle. Jusqu'à ce que toutes les instances de toutes les colonnes correspondantes soient configurées de façon appropriée, la valeur de l'instance correspondante de la colonne vacmSecurityToGroupStatus est 'notReady'. En particulier, une rangée nouvellement créée ne peut pas être rendue active tant qu'une valeur n'a pas été établie pour vacmGroupName. La convention textuelle RowStatus [RFC2579] exige que cette clause DESCRIPTION déclare dans quelles circonstances d'autres objets de cette rangée peuvent être modifiés : la valeur de cet objet n'a pas d'effet sur la possibilité de modification d'autres objets de cette rangée conceptuelle."
::= { vacmSecurityToGroupEntry 5 }

-- Information sur les droits d'accès **********

TYPE D'OBJET vacmAccessTable
SYNTAXE : SEQUENCE DE VacmAccessEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Tableau des droits d'accès pour les groupes. Chaque entrée est indexée par un groupName, un contextPrefix, un securityModel et un securityLevel. Pour déterminer si l'accès est permis, une entrée de ce tableau doit être choisie et le viewName approprié de cette entrée doit être utilisé pour la vérification du contrôle d'accès.

Pour choisir l'entrée appropriée, on suit ces étapes :
1) L'ensemble des correspondances possibles est formé par l'intersection des ensembles d'entrées suivants :
- l'ensemble des entrées qui ont un vacmGroupName identique
- l'union de ces deux ensembles :
  - l'ensemble des entrées qui ont un vacmAccessContextPrefix identique
  - l'ensemble des entrées qui ont une valeur vacmAccessContextMatch de 'prefix' et correspondant à vacmAccessContextPrefix

  entrecoupé par l'union de ces deux ensembles :
  - l'ensemble des entrées qui ont un vacmSecurityModel identique
  - l'ensemble des entrées qui ont une valeur vacmSecurityModel de 'any' entrecoupé par l'ensemble des entrées qui ont une valeur vacmAccessSecurityLevel inférieure ou égale au securityLevel demandé.

2) Si cet ensemble a seulement un membre, c'est fini, autrement, cela revient à décider comment pondérer la préférence entre ContextPrefix, SecurityModel, et SecurityLevel comme suit :
   a) si le sous ensemble d'entrées avec le securityModel correspondant au securityModel dans le message n'est pas vide, on élimine le reste ;
   b) si le sous ensemble d'entrées avec vacmAccessContextPrefix correspondant à contextName dans le message n'est pas vide, on éliminer le reste ;
   c) on élimine toutes les entrées avec ContextPrefix plus court que le plus long qui reste dans l'ensemble.
   d) on choisit l'entrée qui le plus fort securityLevel.

On notera que pour le niveau de sécurité noAuthNoPriv, tous les groupes sont réellement équivalents car l'hypothèse que le securityName a été authentifié ne tient pas. "
::= { vacmMIBObjects 4 }

TYPE D'OBJET vacmAccessEntry
SYNTAXE : VacmAccessEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Droit d'accès configuré dans le LCD qui autorise l'accès à un contexte SNMP. Les entrées dans ce tableau peuvent utiliser une valeur d'instance pour l'objet vacmGroupName même si aucune entrée dans le tableau vacmAccessSecurityToGroupTable n'a de valeur correspondante pour l'objet vacmGroupName. "
INDICE : { vacmGroupName, vacmAccessContextPrefix, vacmAccessSecurityModel, vacmAccessSecurityLevel }
::= { vacmAccessTable 1 }

VacmAccessEntry ::= SEQUENCE
{

vacmAccessContextPrefix	SnmpAdminString, *(chaîne administrative SNMP)*
vacmAccessSecurityModel	SnmpSecurityModel, *(modèle de sécurité SNMP)*
vacmAccessSecurityLevel	SnmpSecurityLevel, *(niveau de sécurité SNMP)*
vacmAccessContextMatch	ENTIER,
vacmAccessReadViewName	SnmpAdminString,
vacmAccessWriteViewName	SnmpAdminString,
vacmAccessNotifyViewName	SnmpAdminString,
vacmAccessStorageType	StorageType, *(type de mémorisation)*
vacmAccessStatus	RowStatus *(statut de rangée)*

}

TYPE D'OBJET vacmAccessContextPrefix
SYNTAXE : SnmpAdminString (TAILLE(0..32))
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Pour obtenir les droits d'accès permis par cette rangée conceptuelle, un contextName doit correspondre exactement (si la valeur de vacmAccessContextMatch est 'exact') ou partiellement (si la valeur de vacmAccessContextMatch est 'prefix') à la valeur de l'instance de cet objet. "
::= { vacmAccessEntry 1 }

TYPE D'OBJET vacmAccessSecurityModel
SYNTAXE : SnmpSecurityModel
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Pour obtenir les droits d'accès permis par cette rangée conceptuelle, ce securityModel doit être utilisé. "

::= { vacmAccessEntry 2 }

TYPE D'OBJET vacmAccessSecurityLevel
SYNTAXE : :SnmpSecurityLevel
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Niveau minimum de sécurité exigé pour obtenir les droits d'accès permis par cette rangée conceptuelle. Un securityLevel de noAuthNoPriv est moins que authNoPriv qui à son tour est moins que authPriv. Si plusieurs entrées sont également indexées sauf pour cet indice vacmAccessSecurityLevel, l'entrée qui a la valeur la plus forte pour vacmAccessSecurityLevel est retenue. "
::= { vacmAccessEntry 3 }

TYPE D'OBJET vacmAccessContextMatch
SYNTAXE : ENTIER { exact (1), -- correspondance exacte du préfixe et du contextName
prefix (2) – correspond seulement au préfixe }
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Si la valeur de cet objet est exact(1), alors toutes les rangées où le contextName correspond exactement à vacmAccessContextPrefix sont retenues. Si la valeur de cet objet est prefix(2), alors toutes les rangées où le contextName dont les octets de début correspondent exactement à vacmAccessContextPrefix sont retenues. Cela permet une forme simple de caractère générique. "
DEFVAL : { exact }
::= { vacmAccessEntry 4 }

TYPE D'OBJET vacmAccessReadViewName
SYNTAXE : SnmpAdminString (TAILLE(0..32))
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "La valeur d'une instance de cet objet identifie la vue de MIB du contexte SNMP auquel cette rangée conceptuelle autorise l'accès en lecture. La vue de MIB identifiée est celle pour laquelle le vacmViewTreeFamilyViewName a la même valeur que l'instance de cet objet ; si la valeur est la chaîne vide, ou si il n'y a pas de vue de MIB active ayant cette valeur de vacmViewTreeFamilyViewName, aucun accès n'est alors accordé. "
DEFVAL : { "H } -- la chaîne vide
::= { vacmAccessEntry 5 }

TYPE D'OBJET vacmAccessWriteViewName
SYNTAXE : SnmpAdminString (TAILLE(0..32))
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "La valeur d'une instance de cet objet identifie la vue de MIB du contexte SNMP auquel cette rangée conceptuelle autorise l'accès en écriture. La vue de MIB identifiée est celle pour laquelle le vacmViewTreeFamilyViewName a la même valeur que l'instance de cet objet ; si la valeur est la chaîne vide, ou si il n'y a pas de vue de MIB active qui a cette valeur de vacmViewTreeFamilyViewName, aucun accès n'est accordé. "
DEFVAL : { "H } -- la chaîne vide
::= { vacmAccessEntry 6 }

TYPE D'OBJET vacmAccessNotifyViewName
SYNTAXE : SnmpAdminString (TAILLE(0..32))
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "La valeur d'une instance de cet objet identifie la vue de MIB du contexte SNMP auquel cette rangée conceptuelle autorise l'accès pou les notifications. La vue de MIB identifiée est celle pour laquelle le vacmViewTreeFamilyViewName a la même valeur que l'instance de cet objet ; si la valeur est la chaîne vide ou si il n'y a pas de vue de MIB active ayant cette valeur de vacmViewTreeFamilyViewName, aucun accès n'est accordé. "
DEFVAL : { "H } -- la chaîne vide
::= { vacmAccessEntry 7 }

TYPE D'OBJET vacmAccessStorageType
SYNTAXE : StorageType
MAX-ACCESS : lecture-création

STATUT : actuel
DESCRIPTION : "Type de mémorisation pour cette rangée conceptuelle. Les rangées conceptuelles qui ont la valeur de 'permanent' n'ont pas besoin de permettre l'accès en écriture à des objets de colonnes de cette rangée. "
DEFVAL : { nonVolatile }
::= { vacmAccessEntry 8 }

TYPE D'OBJET vacmAccessStatus
SYNTAXE : RowStatus
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Statut de cette rangée conceptuelle. La convention textuelle RowStatus [RFC2579] exige que cette clause DESCRIPTION déclare dans quelles circonstances les autres objets de cette rangée peuvent être modifiés : la valeur de cet objet n'a pas d'effet sur la possibilité de modification d'autres objets de cette rangée conceptuelle. "
::= { vacmAccessEntry 9 }

-- Informations sur les vues de MIB *

-- La prise en charge de la granularité de niveau instance est facultative. Dans certaines mises en œuvre, la granularité du contrôle d'accès au niveau de l'instance peut avoir un coût de performance élevé. Les gestionnaires devraient éviter de demander une telle configuration sans nécessité.

vacmMIBViews IDENTIFIANT D'OBJET ::= { vacmMIBObjects 5 }

TYPE D'OBJET vacmViewSpinLock
SYNTAXE : TestAndIncr
MAX-ACCESS : read-write
STATUT : actuel
DESCRIPTION : "Verrou facultatif utilisé pour permettre aux applications de générateur de commandes SNMP coopérantes de coordonner leur utilisation de l'opération Set lors de la création ou la modification de vues. À la création d'une nouvelle vue ou lors de l'altération d'une vue existante, il est important de comprendre les interactions potentielles avec les autres utilisations de la vue. Le vacmViewSpinLock devrait être sauvegardé. Le nom de la vue à créer devrait être déterminé comme unique par l'application de générateur de commandes SNMP en consultant le tableau vacmViewTreeFamilyTable. Finalement, la vue désignée peut être créée (Set), en incluant le verrou facultatif. Si une autre application de générateur de commandes SNMP a altéré les vues dans l'intervalle, la valeur du verrou rotatif aura alors changé, et cette création va donc échouer parce qu'elle va spécifier une mauvaise valeur pour le verrou. Comme c'est un verrou facultatif, l'utilisation de ce verrou n'est pas obligatoire. "
::= { vacmMIBViews 1 }

TYPE D'OBJET vacmViewTreeFamilyTable
SYNTAXE : SEQUENCE DE VacmViewTreeFamilyEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Informations détenues en local sur les familles de sous arborescences au sein des vues de MIB. Chaque vue de MIB est définie par deux ensembles de sous arborescences de vues : les sous arborescences de vues incluses, et les sous arborescences de vues exclues. Chacune de ces sous arborescences de vues, aussi bien incluses qu'exclues, est définie dans ce tableau. Pour déterminer si une instance d'objet particulière est dans une certaine vue de MIB, on compare l'IDENTIFIANT D'OBJET de l'instance d'objet à chacune des entrées actives de la vue de MIB du tableau. Si aucune ne correspond, l'instance d'objet n'est alors pas dans la vue de MIB. Si une ou plusieurs correspondent, l'instance d'objet est alors incluse, ou exclue, de la vue de MIB selon la valeur du vacmViewTreeFamilyType dans l'entrée dont la valeur de vacmViewTreeFamilySubtree a le plus de sous identifiants. Si plusieurs entrées correspondent et ont le même nombre de sous identifiants (lorsque le caractère générique est spécifié avec la valeur de vacmViewTreeFamilyMask) alors l'instance la plus grande lexicographiquement de vacmViewTreeFamilyType détermine l'inclusion ou l'exclusion. Un IDENTIFIANT D'OBJET X d'instance d'objet correspond à une entrée active dans ce tableau lorsque le nombre de sous identifiants dans X est au moins autant que dans la valeur de vacmViewTreeFamilySubtree pour l'entrée, et chaque sous identifiant dans la valeur de vacmViewTreeFamilySubtree correspond à son sous identifiant correspondant dans X. Deux sous identifiants correspondent si le bit correspondant de la valeur de vacmViewTreeFamilyMask pour l'entrée est zéro (la valeur de 'caractère générique') ou si ils sont égaux. Une 'famille' de sous arborescences est l'ensemble de sous arborescences défini par une combinaison particulière de valeurs de vacmViewTreeFamilySubtree et de vacmViewTreeFamilyMask. Dans le cas où aucun 'caractère générique' n'est défini dans le

vacmViewTreeFamilyMask, la famille de sous arborescences se réduit à une seule sous arborescence. Lors de la création ou du changement des vues de MIB, une application SNMP de générateur de commandes devrait utiliser le vacmViewSpinLock pour essayer d'éviter les collisions. Voir la clause DESCRIPTION de vacmViewSpinLock.

Lors de la création de vues de MIB, il est fortement recommandé que les vacmViewTreeFamilyEntries 'exclues' soient créées en premier et ensuite les entrées 'includes'.

Lors d'une suppression de vue de MIB, il est fortement recommandé de supprimer d'abord les vacmViewTreeFamilyEntries 'incluses' et ensuite les entrées 'exclues'.

Si une création pour une entrée de contrôle d'accès de niveau instance est reçue et que la mise en œuvre ne prend pas en charge la granularité de niveau instance, une erreur "inconsistentName" doit être retournée. "

::= { vacmMIBViews 2 }

TYPE D'OBJET vacmViewTreeFamilyEntry
SYNTAXE : VacmViewTreeFamilyEntry
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Informations sur une famille particulière de sous arborescences de vues incluses ou exclues d'une vue de MIB d'un contexte SNMP particulier.

Les mises en œuvre ne doivent pas restreindre le nombre de familles de sous arborescences de vues pour une vue de MIB donnée, sauf comme imposé par les contraintes de ressources sur le nombre global d'entrées dans le tableau vacmViewTreeFamilyTable.

Si il n'existe aucune rangée conceptuelle dans ce tableau pour une certaine vue de MIB (viewName), cette vue peut être considérée comme consistant en l'ensemble vide de sous arborescences de vues. "
INDICE : { vacmViewTreeFamilyViewName, vacmViewTreeFamilySubtree }
::= { vacmViewTreeFamilyTable 1 }

VacmViewTreeFamilyEntry ::= SEQUENCE
{
vacmViewTreeFamilyViewName SnmpAdminString,
vacmViewTreeFamilySubtree IDENTIFIANT D'OBJET,
vacmViewTreeFamilyMask CHAINE D'OCTETS,
vacmViewTreeFamilyType ENTIER,
vacmViewTreeFamilyStorageType StorageType,
vacmViewTreeFamilyStatus RowStatus
}

TYPE D'OBJET vacmViewTreeFamilyViewName
SYNTAXE : SnmpAdminString (TAILLE(1..32))
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Nom lisible par l'homme d'une famille de sous arborescences de vues. "
::= { vacmViewTreeFamilyEntry 1 }

TYPE D'OBJET vacmViewTreeFamilySubtree
SYNTAXE : IDENTIFIANT D'OBJET
MAX-ACCESS : non accessible
STATUT : actuel
DESCRIPTION : "Sous arborescence de MIB qui, lorsque combinée avec l'instance de vacmViewTreeFamilyMask correspondante définit une famille de sous arborescences de vues. "
::= { vacmViewTreeFamilyEntry 2 }

TYPE D'OBJET vacmViewTreeFamilyMask
SYNTAXE : CHAINE D'OCTETS (TAILLE (0..16))
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Gabarit binaire qui, en combinaison avec l'instance correspondante de vacmViewTreeFamilySubtree, définit une famille de sous arborescences de vues.

Chaque bit de ce gabarit binaire correspond à un sous identifiant de vacmViewTreeFamilySubtree, avec le bit de poids fort du ième octet de cette valeur de chaîne d'octets (étendue si nécessaire, voir ci-dessous) correspondant au (8*i - 7)ème sous identifiant, et le bit de moindre poids du ième octet de cette chaîne d'octets correspondant au (8*i)ème sous identifiant, où i est dans la gamme de 1 à 16.

Chaque bit de ce gabarit binaire spécifie si les sous identifiants correspondants doivent ou non être satisfaits lors de

la détermination de si un IDENTIFIANT D'OBJET est dans cette famille de sous arborescences de vues ; un '1' indique qu'une correspondance exacte doit se produire ; un '0' indique un 'caractère générique', c'est-à-dire que toute valeur de sous identifiant correspond.

Donc, l'IDENTIFIANT D'OBJET X d'une instance d'objet est contenue dans une famille de sous arborescences de vues si, pour chaque sous identifiant de la valeur de vacmViewTreeFamilySubtree, soit le ième bit de vacmViewTreeFamilyMask est 0, soit le ième sous identifiant de X est égal au ième sous identifiant de la valeur de vacmViewTreeFamilySubtree. Si la valeur de ce gabarit binaire est long de M bits et si il y a plus de M sous identifiants dans l'instance correspondante de vacmViewTreeFamilySubtree, alors le gabarit binaire est étendu avec des uns pour avoir la longueur requise.

Noter que lorsque la valeur de cet objet est la chaîne de longueur zéro, cette règle d'extension résulte en l'utilisation d'un gabarit qui est tout de uns (c'est-à-dire, pas de 'caractère générique') et la famille de sous arborescences de vues est la sous arborescence de vues identifiée de façon univoque par l'instance correspondante de vacmViewTreeFamilySubtree.

Noter que les gabarits de longueur supérieure à zéro n'ont pas besoin d'être pris en charge. Dans ce cas, cet objet est en lecture seule. "

DEFVAL : { "H }
::= { vacmViewTreeFamilyEntry 3 }

TYPE D'OBJET vacmViewTreeFamilyType
SYNTAXE : ENTIER { included(1), excluded(2) }
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Indique si l'instance correspondante de vacmViewTreeFamilySubtree et vacmViewTreeFamilyMask définit une famille de sous arborescences de vues qui est incluse ou exclue de la vue de MIB. "
DEFVAL : { incluse }
::= { vacmViewTreeFamilyEntry 4 }

TYPE D'OBJET vacmViewTreeFamilyStorageType
SYNTAXE : StorageType
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Type de mémorisation pour cette rangée conceptuelle. Les rangées conceptuelles qui ont la valeur 'permanent' n'ont pas besoin de permettre l'accès en écriture aux objets des colonnes de cette rangée. "
DEFVAL : { nonVolatile }
::= { vacmViewTreeFamilyEntry 5 }

TYPE D'OBJET vacmViewTreeFamilyStatus
SYNTAXE : RowStatus
MAX-ACCESS : lecture-création
STATUT : actuel
DESCRIPTION : "Statut de cette rangée conceptuelle. La valeur de cet objet n'a pas d'effet sur la possibilité de modification des autres objets de cette rangée conceptuelle. "
::= { vacmViewTreeFamilyEntry 6 }

-- Informations de conformité **********

IDENTIFIANT D'OBJET vacmMIBCompliances ::= { vacmMIBConformance 1 }
IDENTIFIANT D'OBJET vacmMIBGroups ::= { vacmMIBConformance 2 }

-- Déclarations de conformité **********

CONFORMITÉ DE MODULE vacmMIBCompliance
STATUT : actuel
DESCRIPTION : "Déclaration de conformité pour les moteurs SNMP qui mettent en œuvre la MIB de configuration de modèle SNMP de contrôle d'accès fondé sur la vue. "
MODULE : ce module
GROUPES OBLIGATOIRES : { vacmBasicGroup }

OBJET vacmAccessContextMatch
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmAccessReadViewName
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmAccessWriteViewName
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmAccessNotifyViewName
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmAccessStorageType
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmAccessStatus
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en création/suppression/modification au tableau vacmAccessTable n'est pas exigé. "

OBJET vacmViewTreeFamilyMask
SYNTAXE D'ÉCRITURE : CHAINE D'OCTETS (TAILLE (0))
MIN-ACCESS : lecture seule
DESCRIPTION : "La prise en charge de la configuration via SNMP des familles de sous arborescences en utilisant des caractères génériques n'est pas exigée. "

OBJET vacmViewTreeFamilyType
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmViewTreeFamilyStorageType
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en écriture n'est pas exigé."

OBJET vacmViewTreeFamilyStatus
MIN-ACCESS : lecture seule
DESCRIPTION : "L'accès en création/suppression/modification au tableau vacmViewTreeFamilyTable n'est pas exigé. "
::= { vacmMIBCompliances 1 }

-- **Unités de conformité** **********

GROUPE D'OBJET vacmBasicGroup
OBJETS { vacmContextName, vacmGroupName, vacmSecurityToGroupStorageType, vacmSecurityToGroupStatus, vacmAccessContextMatch, vacmAccessReadViewName, vacmAccessWriteViewName, vacmAccessNotifyViewName, vacmAccessStorageType, vacmAccessStatus, vacmViewSpinLock, vacmViewTreeFamilyMask, vacmViewTreeFamilyType, vacmViewTreeFamilyStorageType, vacmViewTreeFamilyStatus }
STATUT : actuel
DESCRIPTION : "Collection d'objets qui assurent la configuration à distance d'un moteur SNMP mettant en œuvre le modèle SNMP de contrôle d'accès fondé sur la vue. "
::= { vacmMIBGroups 1 }

FIN

5. Propriété intellectuelle

L'IETF ne prend pas position sur la validité et la portée de tout droit de propriété intellectuelle ou autres droits qui pourraient être revendiqués au titre de la mise en œuvre ou l'utilisation de la technologie décrite dans le présent document ou sur la mesure dans laquelle toute licence sur de tels droits pourrait être ou n'être pas disponible ; pas plus qu'elle ne

prétend avoir accompli aucun effort pour identifier de tels droits. Les informations sur les procédures de l'ISOC au sujet des droits dans les documents de l'ISOC figurent dans les BCP 78 et BCP 79.

Des copies des dépôts d'IPR faites au secrétariat de l'IETF et toutes assurances de disponibilité de licences, ou le résultat de tentatives faites pour obtenir une licence ou permission générale d'utilisation de tels droits de propriété par ceux qui mettent en œuvre ou utilisent la présente spécification peuvent être obtenues sur le répertoire en ligne des IPR de l'IETF à http://www.ietf.org/ipr.

L'IETF invite toute partie intéressée à porter son attention sur tous copyrights, licences ou applications de licence, ou autres droits de propriété qui pourraient couvrir les technologies qui peuvent être nécessaires pour mettre en œuvre la présente norme. Prière d'adresser les informations à l'IETF à ietf- ipr@ietf.org.

6. Remerciements

Le présent document est le résultat des efforts du groupe de travail SNMPv3. Des remerciements particuliers sont adressés par ordre alphabétique aux membres suivants du GT SNMPv3 : Harald Tveit Alvestrand (Maxware), Dave Battle (SNMP Research, Inc.), Alan Beard (Disney Worldwide Services), Paul Berrevoets (SWI Systemware/Halcyon Inc.), Martin Bjorklund (Ericsson), Uri Blumenthal (IBM T.J. Watson Research Center), Jeff Case (SNMP Research, Inc.), John Curran (BBN), Mike Daniele (Compaq Computer Corporation), T. Max Devlin (Eltrax Systems), John Flick (Hewlett Packard), Rob Frye (MCI), Wes Hardaker (U.C.Davis, Information Technology - D.C.A.S.), David Harrington (Cabletron Systems Inc.), Lauren Heintz (BMC Software, Inc.), N.C. Hien (IBM T.J. Watson Research Center), Michael Kirkham (InterWorking Labs, Inc.), Dave Levi (SNMP Research, Inc.), Louis A Mamakos (UUNET Technologies Inc.), Joe Marzot (Nortel Networks), Paul Meyer (Secure Computing Corporation), Keith McCloghrie (Cisco Systems), Bob Moore (IBM), Russ Mundy (TIS Labs at Network Associates), Bob Natale (ACE*COMM Corporation), Mike O'Dell (UUNET Technologies Inc.), Dave Perkins (DeskTalk), Peter Polkinghorne (Brunel University), Randy Presuhn (BMC Software, Inc.), David Reeder (TIS Labs at Network Associates), David Reid (SNMP Research, Inc.), Aleksey Romanov (Quality Quorum), Shawn Routhier (Epilogue), Juergen Schoenwaelder (TU Braunschweig), Bob Stewart (Cisco Systems), Mike Thatcher (Independent Consultant), Bert Wijnen (IBM T.J. Watson Research Center).

Le document se fonde sur les recommandations de l'équipe conseil Évolution du cadre administratif et de sécurité pour SNMP de l'IETF. Les membres de cette équipe conseil étaient : David Harrington (Cabletron Systems Inc.), Jeff Johnson (Cisco Systems), David Levi (SNMP Research Inc.), John Linn (Openvision), Russ Mundy (Trusted Information Systems) (président), Shawn Routhier (Epilogue), Glenn Waters (Nortel), Bert Wijnen (IBM T. J. Watson Research Center).

Comme recommandé par l'équipe conseil et la charte du groupe de travail SNMPv3, la conception a incorporé autant que faire s'est pu des précédentes RFC et projets. Il en résulte que des remerciement particuliers sont dus aux auteurs des projets précédents connus sous les noms de SNMPv2u et de SNMPv2* : Jeff Case (SNMP Research, Inc.), David Harrington (Cabletron Systems Inc.), David Levi (SNMP Research, Inc.), Keith McCloghrie (Cisco Systems), Brian O'Keefe (Hewlett Packard), Marshall T. Rose (Dover Beach Consulting), Jon Saperia (BGS Systems Inc.), Steve Waldbusser (International Network Services), Glenn W. Waters (Bell-Northern Research Ltd.).

7. Considérations pour la sécurité

7.1 Pratiques recommandées

Le présent document est destiné à être utilisé dans l'architecture SNMP. Le modèle de contrôle d'accès fondé sur la vue décrit dans le présent document vérifie les droits d'accès aux informations de gestion sur la base :

- du contextName *(nom de contexte)* représentant un ensemble d'informations de gestion au système géré où le module de contrôle d'accès fonctionne.
- du groupName *(nom de groupe)* représentant un ensemble de zéro, un ou plusieurs securityName. La combinaison d'un securityModel et d'un securityName est transposée en un groupe dans le modèle de contrôle d'accès fondé sur la vue.
- du securityModel *(modèle de sécurité)* sous lequel l'accès est demandé.
- du securityLevel *(niveau de sécurité)* sous lequel l'accès est demandé.
- de l'opération effectuée sur les informations de gestion.
- des vues de MIB pour l'accès en lecture, écriture ou notification.

Lorsque le module de contrôle d'accès fondé sur l'utilisateur est invoqué pour vérifier les droits d'accès, on suppose que le module appelant s'est assuré des aspects d'authentification et de confidentialité comme spécifié par le securityLevel qui est communiqué.

Lors de la création ou la suppression d'entrées du tableau vacmViewTreeFamilyTable, il est important de faire dans la séquence des opérations comme recommandé dans la clause DESCRIPTION de la définition de vacmViewTreeFamilyTable. Autrement, un accès non désiré peut être accordé lors du changement des entrées dans le tableau.

7.2 Définition des groupes

Les noms de groupe (groupName) sont utilisés pour donner l'accès à un groupe de zéro, un ou plusieurs noms de sécurité (securityName). Au sein du modèle de contrôle d'accès fondé sur la vue, un groupName est considéré comme existant si ce groupName est cité dans le tableau vacmSecurityToGroupTable.

En transposant la combinaison d'un modèle de sécurité (securityModel) et d'un nom de sécurité (securityName) en un nom de groupe (groupName), une application SNMP de générateur de commandes peut ajouter/supprimer des noms de sécurité dans un groupe, si l'accès approprié est permis.

De plus, il est important de réaliser que le groupement de paires <securityModel, securityName> dans le tableau vacmSecurityToGroupTable ne prend pas en compte le niveau de sécurité. Il est donc important que l'administrateur de la sécurité utilise l'indice de niveau de sécurité (securityLevel) dans vacmAccessTable pour séparer les accès noAuthNoPriv des accès authPriv et/ou authNoPriv.

7.3 Conformité

Pour qu'une mise en œuvre du modèle de contrôle d'accès fondé sur la vue soit conforme, elle DOIT mettre en œuvre la MIB SNMP-VIEW-BASED-ACM conformément à vacmMIBCompliance. Elle DEVRAIT aussi mettre en œuvre la configuration initiale, décrite à l'Appendice A.

7.4 Accès à la MIB SNMP-VIEW-BASED-ACM

Les objets de cette MIB contrôlent l'accès à toutes les données de MIB qui sont accessibles via le moteur SNMP et qui peuvent être considérés comme sensibles dans de nombreux environnements. Il est important de contrôler étroitement l'accès (en lecture et en écriture) à ces objets de MIB en utilisant des modèles de contrôle d'accès configurés de façon appropriée (par exemple, le modèle de contrôle d'accès fondé sur la vue spécifié dans le présent document).

8. Références

8.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997.

[RFC2578] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure des informations de gestion, version 2 (SMIv2)", avril 1999. (STD0058)

[RFC2579] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Conventions textuelles pour SMIv2", avril 1999. (STD0058)

[RFC2580] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Déclarations de conformité pour SMIv2", avril 1999. (STD0058)

[RFC3411] D. Harrington, R. Presuhn, B. Wijnen, "Architecture de description des cadres de gestion du protocole simple de gestion de réseau (SNMP)", décembre 2002. (*MàJ par* RFC5343) (STD0062)

[RFC3412] J. Case et autres, "Traitement et distribution de message pour le protocole simple de gestion de réseau (SNMP)", décembre 2002. (STD0062)

[RFC3414] U. Blumenthal, B. Wijnen, "Modèle de sécurité fondée sur l'utilisateur (USM) pour la version 3 du protocole simple de gestion de réseau (SNMPv3)", décembre 2002. (STD0062)

8.2 Références pour information

[ISO-ASN.1] Organisation Internationale de Normalisation, Norme Internationale 8824 "Systèmes de traitement de l'information - Interconnexion des systèmes ouverts - Spécification de la notation de syntaxe abstraite numéro un (ASN.1)", décembre 1987.

Appendice A Installation

A.1 Paramètres d'installation

Durant l'installation, un moteur SNMP d'autorité qui prend en charge le présent modèle de contrôle d'accès fondé sur la vue DEVRAIT être configuré avec plusieurs paramètres initiaux. Pour le modèle de contrôle d'accès fondé sur la vue, cela inclut :

1) Une configuration de sécurité
Le choix de la configuration de sécurité détermine si la configuration initiale est mise en œuvre, et comment elle l'est. Un des trois choix possibles est retenu :
- initial-minimum-security-configuration *(configuration initiale de sécurité minimum)*
- initial-semi-security-configuration *(configuration initiale de semi sécurité)*
- initial-no-access-configuration *(configuration initiale d'accès interdit)*

Dans le cas d'une initial-no-access-configuration, il n'y a pas de configuration initiale, et donc les étapes suivantes ne sont pas applicables.

2) Un contexte par défaut
Une entrée dans le vacmContextTable avec un contextName de "" (la chaîne vide) représentant le contexte par défaut. Noter que ce tableau est créé automatiquement si il existe un contexte par défaut.

vacmContextName	""

3) Un groupe initial
Une entrée dans le vacmSecurityToGroupTable pour permettre l'accès au groupe "initial".

vacmSecurityModel	3 (USM)
vacmSecurityName	"initial"
vacmGroupName	"initial"
vacmSecurityToGroupStorageType	anyValidStorageType
vacmSecurityToGroupStatus	actif

4) Droits d'accès initiaux
Trois entrées dans le vacmAccessTable comme suit :
- accès read-notify *(lecture-notification)* pour le modèle de sécurité USM, niveau de sécurité de "noAuthNoPriv" au nom des noms de sécurité qui appartiennent au groupe "initial" de la vue de MIB <restricted> dans le contexte par défaut avec le contextName "".
- read-write-notify *(lecture-écriture-notification)* pour le modèle de sécurité USM, niveau de sécurité de "authNoPriv" au nom des noms de sécurité qui appartiennent au groupe "initial" de la vue de MIB <internet> dans le contexte par défaut avec le contextName "".
- si la confidentialité est prise en charge, l'accès en lecture écriture notification pour le modèle de sécurité USM, au niveau de sécurité "authPriv" au nom des noms de sécurité qui appartiennent au groupe "initial" de la vue de MIB <internet> dans le contexte par défaut avec le contextName "".

Cela se traduit dans les entrées suivantes du vacmAccessTable.

- Une entrée à utiliser pour l'accès non authentifié (noAuthNoPriv) :

vacmGroupName	"initial"
vacmAccessContextPrefix	""
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	noAuthNoPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	"restricted"
vacmAccessWriteViewName	""
vacmAccessNotifyViewName	"restricted"
vacmAccessStorageType	anyValidStorageType

vacmAccessStatus	active

- Une entrée à utiliser pour l'accès authentifié (authNoPriv) avec confidentialité facultative (authPriv) :

vacmGroupName	"initial"
vacmAccessContextPrefix	""
vacmAccessSecurityModel	3 (USM)
vacmAccessSecurityLevel	authNoPriv
vacmAccessContextMatch	exact
vacmAccessReadViewName	"internet"
vacmAccessWriteViewName	"internet"
vacmAccessNotifyViewName	"internet"
vacmAccessStorageType	anyValidStorageType
vacmAccessStatus	active

5) Deux vues de MIB, dont la seconde dépend de la configuration de sécurité.

- Une vue, la vue <internet>, pour l'accès authentifié :
  - la vue de MIB <internet> est la sous arborescence suivante :
    "internet" (subtree 1.3.6.1)

- Une seconde vue, la vue <restricted>, pour l'accès non authentifié. Cette vue est configurée conformément à la configuration de sécurité choisie :

- Pour l'accès interdit, il n'y a pas de configuration initiale par défaut, aussi aucune vue de MIB n'est prescrite.

- Pour la configuration initiale semi sécurisée :
  la vue de MIB <restricted> est l'union de ces sous arborescences :

(a) "system"	(subtree 1.3.6.1.2.1.1)	[RFC3918]
(b) "snmp"	(subtree 1.3.6.1.2.1.11)	[RFC3918]
(c) "snmpEngine"	(subtree 1.3.6.1.6.3.10.2.1)	[RFC3411]
(d) "snmpMPDStats"	(subtree 1.3.6.1.6.3.11.2.1)	[RFC3412]
(e) "usmStats"	(subtree 1.3.6.1.6.3.15.1.1)	[RFC3414]

- Pour la configuration initiale sécurité minimum :
  la vue de MIB <restricted> est la sous arborescence suivante .
  "internet" (subtree 1.3.6.1)

Ceci se traduit dans l'entrée "internet" suivante dans le vacmViewTreeFamilyTable :

	minimum-secure	semi-secure
vacmViewTreeFamilyViewName	"internet"	"internet"
vacmViewTreeFamilySubtree	1.3.6.1	1.3.6.1
vacmViewTreeFamilyMask	""	""
vacmViewTreeFamilyType	1 (included)	1 (included)
vacmViewTreeFamilyStorageType	anyValidStorageType	anyValidStorageType
vacmViewTreeFamilyStatus	active	active

En plus, cela se traduit dans les entrées "restricted" suivantes dans le vacmViewTreeFamilyTable :

	minimum-secure	semi-secure
vacmViewTreeFamilyViewName	"restricted"	"restricted"
vacmViewTreeFamilySubtree	1.3.6.1	1.3.6.1.2.1.1
vacmViewTreeFamilyMask	""	""
vacmViewTreeFamilyType	1 (included)	1 (included)
vacmViewTreeFamilyStorageType	anyValidStorageType	anyValidStorageType
vacmViewTreeFamilyStatus	active	active
vacmViewTreeFamilyViewName		"restricted"
vacmViewTreeFamilySubtree		1.3.6.1.2.1.11
vacmViewTreeFamilyMask		""
vacmViewTreeFamilyType		1 (included)
vacmViewTreeFamilyStorageType		anyValidStorageType
vacmViewTreeFamilyStatus		active
vacmViewTreeFamilyViewName		"restricted"
vacmViewTreeFamilySubtree		1.3.6.1.6.3.10.2.1

vacmViewTreeFamilyMask	""
vacmViewTreeFamilyType	1 (included)
vacmViewTreeFamilyStorageType	anyValidStorageType
vacmViewTreeFamilyStatus	active
vacmViewTreeFamilyViewName	"restricted"
vacmViewTreeFamilySubtree	1.3.6.1.6.3.11.2.1
vacmViewTreeFamilyMask	""
vacmViewTreeFamilyType	1 (included)
vacmViewTreeFamilyStorageType	anyValidStorageType
vacmViewTreeFamilyStatus	active
vacmViewTreeFamilyViewName	"restricted"
vacmViewTreeFamilySubtree	1.3.6.1.6.3.15.1.1
vacmViewTreeFamilyMask	""
vacmViewTreeFamilyType	1 (included)
vacmViewTreeFamilyStorageType	anyValidStorageType
vacmViewTreeFamilyStatus	active

Appendice B. Liste des modifications

Changements faits par rapport à la RFC 2575 :
- Suppression des références dans le résumé conformément aux lignes directrices de l'éditeur des RFC
- Mise à jour des références

Changements par rapport à la RFC 2275 :
- Ajout de texte à la clause DESCRIPTION : de vacmSecurityToGroupStatus pour préciser dans quelles conditions une entrée dans le vacmSecurityToGroupTable peut être activée.
- Ajout des clauses REVISION à IDENTITÉ DE MODULE
- Précisions au texte de la clause DESCRIPTION : de vacmAccessTable.
- Ajout d'une clause DEFVAL à l'objet vacmAccessContextMatch.
- Ajout d'une colonne manquante dans l'Appendice A et réarrangement pour plus de clarté.
- Correction des OID dans l'Appendice A.
- Utilisation de la terminologie de classe de PDU à la place des types de PDU de la RFC1905.
- Ajout du paragraphe 7.4 sur le contrôle d'accès à la MIB.
- Correction des références aux documents nouveaux ou révisés.
- Correction des des informations de contact de l'éditeur.
- Correction de fautes de frappe.
- Suppression d'une vacmAccesEntry de l'exemple de l'Appendice A.
- Ajout de quelques éclaicissements.
- Mise à jour de la section des remerciements.

Adresse des éditeurs

Bert Wijnen
Lucent Technologies
Schagen 33
3461 GL Linschoten
Netherlands
téléphone : +31-348-480-685
mél : bwijnen@lucent.com

Randy Presuhn
BMC Software, Inc.
2141 North First Street
San Jose, CA 95131
USA
téléphone : +1 408-546-1006
mél : randy_presuhn@bmc.com

Keith McCloghrie
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
téléphone : +1-408-526-5260
mél : kzm@cisco.com

Déclaration de droits de reproduction

nécessaire à l'élaboration des normes Internet, auquel cas les procédures pour les droits de reproduction définis dans les processus de normes pour Internet doivent être suivies, ou si nécessaire pour le traduire dans des langues autres que l'anglais.

Les permissions limitées accordées ci-dessus sont perpétuelles et ne seront pas révoquées par la Société Internet ou ses successeurs ou ayants droit.

Remerciement
Le financement de la fonction d’édition des RFC est actuellement fourni par la Internet Society.