Équipe d'ingénierie de l'Internet (IETF)
Request for Comments : 8341
STD : 91
RFC rendue obsolète : 6536
Catégorie : Norme
ISSN: 2070-1721

A. Bierman, YumaWorks
M. Bjorklund, Tail-f Systems

mars 2018
Traduction Claude Brière de L'Isle

Modèle de contrôle d'accès à la configuration réseau

Résumé
La normalisation des interfaces de configuration réseau à utiliser avec le protocole de configuration de réseau (NETCONF, *Network Configuration Protocol*) ou le protocole RESTCONF exige un environnement de fonctionnement structuré et sûr qui promeuve l'utilisabilité par l'homme et l'interopérabilité multi fabricants. Il est besoin d'un mécanisme standard pour restreindre l'accès des utilisateurs particuliers au protocole NETCONF ou RESTCONF à un sous ensemble préconfiguré de toutes les opérations et contenus disponibles des protocoles NETCONF ou RESTCONF. Le présent document définit un tel modèle de contrôle d'accès.
Le présent document rend obsolète la RFC 6536.

Table des Matières

1. Introduction

Le protocole de configuration de réseau (NETCONF) et le protocole RESTCONF ne fournissent aucun mécanisme standard pour restreindre le fonctionnement et le contenu des protocoles auxquels chaque utilisateur est autorisé à accéder.

Il y a besoin d'une gestion interopérable de l'accès contrôlé aux portions choisies par l'administrateur du contenu NETCONF ou RESTCONF disponible au sein d'un serveur particulier.

Le présent document traite des mécanismes de contrôle d'accès pour les couches de fonctionnement et de contenu de NETCONF, comme défini dans la [RFC6241], et RESTCONF, comme défini dans la [RFC8040]. Il contient trois sections principales :
1. Objectifs de conception du contrôle d'accès
2. Modèle du contrôle d'accès NETCONF (NACM, *NETCONF Access Control Model*)
3. Modèles de données YANG (ietf-netconf-acm.yang)

YANG version 1.1 [RFC7950] ajoute deux nouvelles constructions qui ont besoin d'un traitement de contrôle d'accès particulier. La déclaration "action" est similaire à la déclaration "rpc", excepté qu'elle est située dans un nœud de données. La déclaration "notification" peut aussi être située dans un nœud de données.

1.1 Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119], [RFC8174] quand, et seulement quand ils apparaissent tout en majuscules, comme montré ci-dessus.

Les termes suivants sont définis dans la [RFC8342] et ne sont pas redéfinis ici :
- magasin de données *(datastore)*
- magasin de données de configuration *(configuration datastore)*
- magasin de données de configuration conventionnel *(conventional configuration datastore)*
- magasin de données de configuration candidat *(candidate configuration datastore)*
- magasin de données de configuration en cours *(running configuration datastore)*
- magasin de données de configuration de démarrage *(startup configuration datastore)*
- magasin de données d'état de fonctionnement *(operational state datastore)*
- client
- serveur

Les termes suivants sont définis dans la [RFC6241] et ne sont pas redéfinis ici :
- opération de protocole
- session
- utilisateur

Les termes suivants sont définis dans la [RFC7950] et ne sont pas redéfinis ici :
- action
- nœud de données *(data node)*
- déclaration de définition de données *(data definition statement)*

Les termes suivants sont définis dans la [RFC8040] et ne sont pas redéfinis ici :
- ressource de données

- o ressource de magasin de données
- o ressource de fonctionnement
- o ressource cible

Le terme suivant est défini dans la RFC7230] et n'est pas redéfini ici :
- o URI de demande

Les termes suivants sont utilisés dans le présent document :

contrôle d'accès : dispositif de sécurité fourni par le serveur qui permet à un administrateur de restreindre l'accès à un sous ensemble de toutes les opérations et données du protocole, sur la base de divers critères.

modèle de contrôle d'accès (ACM, *access control model*) : modèle conceptuel utilisé pour configurer et surveiller les procédures de contrôle d'accès désirées par l'administrateur pour appliquer une politique de contrôle d'accès particulière.

règle de contrôle d'accès : critère utilisé pour déterminer si une opération d'accès particulière sera permise ou refusée.

opération d'accès : comment une demande tente d'accéder à un objet conceptuel : "none" *(aucun)*, "read" *(lire)*, "create" *(créer)*, "delete" *(supprimer)*, "update" *(mettre à jour)*, ou "execute" *(exécuter)*.

hiérarchie de nœud de données : hiérarchie des nœuds de données qui identifie le nœud spécifique "d'action" ou de "notification" dans le magasin de données.

session de récupération : session administrative spéciale qui reçoit un accès NETCONF illimité et est exempte de toute application de contrôle d'accès. Le ou les mécanismes utilisés par un serveur pour contrôler et identifier si une session est ou non de récupération sont spécifiques de la mise en œuvre et sortent du domaine d'application de ce document.

accès en écriture : abrégé pour les opérations d'accès "créer", "supprimer" et "mettre à jour".

1.2 Changements depuis la RFC 6536

Les procédures NACM et le modèles des données ont été mis à jour pour la prise en charge de nouvelles capacités de modélisation des données dans la version 1.1 du langage YANG de modélisation des données. Les déclarations "action" et "notification" peuvent être utilisées dans les nœuds de données pour définir des opérations et notifications spécifiques du modèle des données.

Un cas d'utilisation important pour ces nouvelles déclarations YANG est la granularité accrue du contrôle d'accès qui peut être réalisé par dessus les déclarations de niveau supérieur "rpc" et "notification". Les nouvelles déclarations "action" et "notification" sont utilisées au sein des nœuds de données, et l'accès à l'action ou notification peut être restreint aux instances spécifiques de ces nœuds de données.

On a ajouté la prise en charge du protocole RESTCONF. Les opérations RESTCONF sont similaires aux opérations NETCONF, de sorte qu'une simple transposition des procédures et modèle de données NACM existantes est possible.

Le comportement d'accès aux nœuds de données pour les confrontations de chemins ont été précisés pour inclure aussi la confrontation des nœuds descendants du chemin spécifié.

Le comportement des droits d'accès de l'opération <edit-config> a été précisé pour indiquer que l'accès en écriture n'est pas exigé pour les nœuds de données qui sont implicitement modifiés par des effets collatéraux (comme l'évaluation de YANG "when-stmts", ou des nœuds de données implicitement supprimés lors de la création d'un nœud de données sous une branche différente sous un "choice-stmt" YANG).

La section des considérations sur la sécurité a été mise à jour pour se conformer aux lignes directrices de sécurité du module YANG [YANG-SEC]. Noter que le module YANG dans ce document ne définit aucune opération de RPC.

2. Objectif de conception du contrôle d'accès

Cette section documente les objectifs de conception du modèle de contrôle d'accès NETCONF présenté à la Section 3.

2.1 Points de contrôle d'accès

NETCONF permet aux mises en œuvre de serveur d'ajouter de nouvelles opérations de protocole adaptées à leurs besoins, et le langage de modélisation de données YANG prend en charge cette disposition. Ces opérations peuvent être définies dans des modules YANG standard ou propriétaires.

Il n'est pas possible de concevoir un ACM pour NETCONF qui se concentre seulement sur un ensemble statique d'opérations de protocole standard défini par NETCONF lui-même, comme certains autres protocoles. Comme peu d'hypothèses peuvent être faites sur une opération de protocole arbitraire, les composants architecturaux de serveur NETCONF doivent être protégés aux trois points de contrôle conceptuels.

Ces points de contrôle d'accès, décrits par la Figure 1, sont :
opération de protocole : permission d'invoquer des opérations de protocole spécifiques ;
magasin de données : permission de lire et/ou modifier des nœuds de données spécifiques au sein de tout magasin de données ;
notification : permission de recevoir des types d'événement de notification spécifiques.

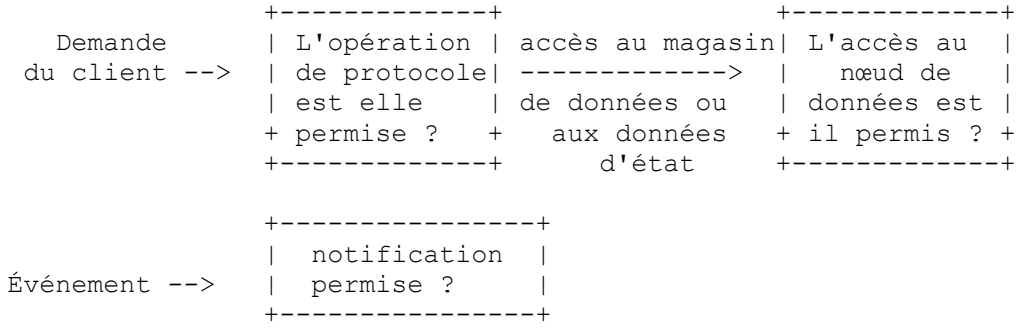


Figure 1

2.2 Simplicité

Il existe une crainte qu'un ACM compliqué ne soit pas largement déployé parce que il serait trop difficile à utiliser. La configuration du système de contrôle d'accès doit être aussi simple que possible. Les tâches simples et courantes doivent être faciles à configurer et exigent peu d'expertise ou de connaissances spécifiques du domaine. Les tâches complexes sont possibles en utilisant des mécanismes supplémentaires qui peuvent exiger plus d'expertise.

Un seul ensemble de règles de contrôle d'accès devait être capable de contrôler tous les types d'invocation d'opération de protocole NETCONF, tous les accès de magasin de données, et tous les événements de notification.

Le contrôle d'accès devrait être défini avec un ensemble de permissions petit et familier, tout en permettant quand même un plein contrôle de l'accès au magasin de données.

2.3 Interface de procédure

NETCONF utilise un modèle d'appel de procédure à distance (RPC, *Remote Procedure Call*) et un ensemble extensible d'opérations de protocole. Le contrôle d'accès pour toute opération de protocole possible est nécessaire.

2.4 Accès au magasin de données

Il est nécessaire de contrôler l'accès aux nœuds et sous arborescences spécifiques au sein du magasin de données, sans considération de l'opération de protocole – standard ou propriétaire – qui a été utilisée pour accéder au magasin de données.

2.5 Utilisateurs et groupes

Il est nécessaire que les règles de contrôle d'accès pour un seul utilisateur ou un groupe configurable d'utilisateurs puissent être configurées.

L'ACM doit prendre en charge le concept de groupe administratif, prendre en charge la distinction bien établie entre un compte racine et les autres types conceptuels de compte d'utilisateur moins privilégié. Ces groupes doivent pouvoir être configurés par l'administrateur.

Il est nécessaire que la transposition d'utilisateur à groupe puisse être déléguée à un serveur central, comme un serveur RADIUS [RFC2865], [RFC5607]. Comme l'authentification est effectuée par la couche transport et que RADIUS effectue l'authentification et l'autorisation de service en même temps, le protocole de transport sous-jacent doit être capable de faire rapport au serveur d'un ensemble de noms de groupes associés à l'utilisateur. Il est nécessaire que l'administrateur puisse désactiver l'usage de ces noms de groupe au sein de l'ACM.

2.6 Maintenance

Il devrait être possible de désactiver une partie ou la totalité des procédures d'application du modèle de contrôle d'accès sans supprimer de règles de contrôle d'accès.

2.7 Capacités de configuration

Des mécanismes convenables de configuration et de surveillance sont nécessaires pour permettre à un administrateur de gérer facilement tous les aspects du comportement de l'ACM. Un modèle de données standard convenant à l'utilisation avec l'opération de protocole <edit-config> devrait être disponible à cette fin.

Les règles de contrôle d'accès pour interdire les opérations d'accès sur des sous arborescences spécifiques au sein du magasin de données de configuration doivent être prises en charge.

2.8 Identification de contenu sensible pour la sécurité

Un des plus importants aspects de la documentation du modèle de données, et un des plus gros soucis durant le déploiement, est l'identification du contenu sensible pour la sécurité. Cela s'applique aux opérations de protocole dans NETCONF, pas seulement aux données et aux notifications.

Il est obligatoire que les objets sensibles pour la sécurité soient documentés dans la section "Considérations pour la sécurité" d'une RFC. C'est bien, mais ce n'est pas assez, pour les raisons suivantes :

- Cette approche par la seule documentation force les administrateurs à étudier la RFC et à déterminer si il y a des risques potentiels de sécurité introduits pas un nouveau modèle de données.
- Si des risques sont identifiés pour la sécurité, l'administrateur doit alors étudier d'autres textes de RFC et déterminer comment atténuer le ou les risques de sécurité.
- L'ACM sur chaque serveur doit être configuré pour atténuer les risques pour la sécurité, par exemple, exiger un accès privilégié pour lire ou écrire les données spécifiques identifiées dans la section "Considérations sur la sécurité".
- Si l'ACM n'est pas préconfiguré, il y aura alors une fenêtre de vulnérabilité après le chargement du nouveau modèle de données et avant que les nouvelles règles de contrôle d'accès pour ce modèle de données soient configurées, activées et déboguées.

Souvent, l'administrateur veut juste désactiver l'accès par défaut au contenu sûr afin qu'aucun changement par inadvertance ou malveillance ne puisse être fait au serveur. Cela permet que les règles par défaut soient plus indulgentes, sans augmenter significativement le risque.

Un concepteur de modèle de données doit être capable d'utiliser des déclaration en langage machine pour identifier le contenu qui doit être protégé par défaut. Cela va permettre que des outils de client et de serveur identifient automatiquement des risques de sécurité spécifiques du modèle de données, en refusant l'accès aux données sensibles sauf si l'utilisateur est explicitement autorisé à effectuer l'opération d'accès demandée.

3. Modèle de contrôle d'accès NETCONF (NACM)

3.1 Généralités

Cette section donne une vue d'ensemble de la structure du modèle de contrôle d'accès. Elle décrit le modèle de traitement du message de protocole NETCONF et les exigences conceptuelles du contrôle d'accès au sein de ce modèle.

3.1.1 Caractéristiques

Le modèle de données NACM fournit les caractéristiques suivantes :

- Contrôle indépendant du RPC, de l'action, des données, et de la notification d'accès.
- Prise en charge du concept de récupération d'urgence de session, mais la configuration du serveur à cette fin sort du

domaine d'application de ce document. Une récupération d'urgence de session va outrepasser toutes les applications de contrôle d'accès, afin de lui permettre d'initialiser ou réparer la configuration NACM.

- Un ensemble simple et familier de permissions de magasin de données est utilisé.
- Prise en charge de l'étiquetage de sécurité YANG (par exemple, une déclaration "nacm:default-deny-write") permet aux modes de sécurité par défaut d'exclure automatiquement les données sensibles.
- Des modes d'accès par défaut séparés pour les permissions d'écriture, de lecture, et d'exécution.
- Des règles de contrôle d'accès sont appliquées aux groupes d'utilisateurs configurables.
- Les procédures d'application du contrôle d'accès peuvent être désactivées durant le fonctionnement, sans supprimer de règles de contrôle d'accès, afin de déboguer des problèmes de fonctionnement.
- Le nombre de demandes d'opération de protocole refusées et de demandes d'écriture de magasin de données refusées peut être surveillé par le client.
- Des identifiants d'instance YANG simples sans contraintes sont utilisés pour configurer les règles de contrôle d'accès pour des nœuds de données spécifiques.

3.1.2 Dépendances externes

NETCONF [RFC6241] et RESTCONF [RFC8040] sont utilisés pour les besoins de la gestion de réseau au sein de ce document.

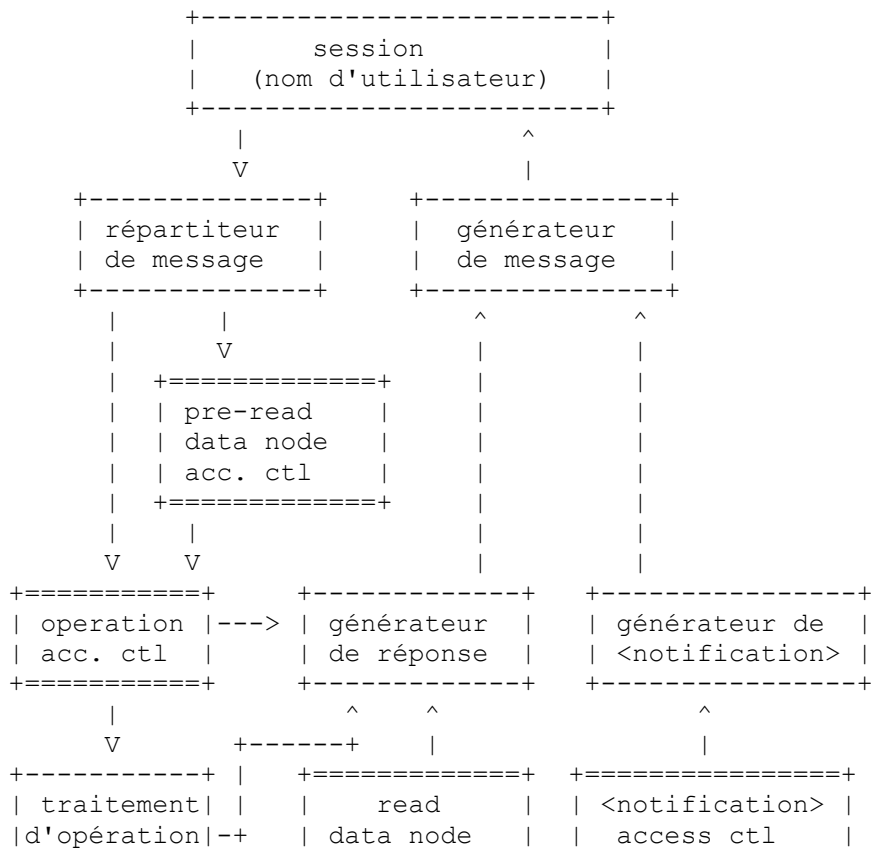
Le langage de modélisation de données YANG [RFC7950] est utilisé pour définir les modèles de données à utiliser avec NETCONF ou RESTCONF. YANG est aussi utilisé pour définir le modèle de données dans ce document.

3.1.3 Modèle de traitement de message

Le diagramme qui suit montre le modèle conceptuel de flux de messages, incluant les points auxquels le contrôle d'accès est appliqué durant le traitement de message NETCONF.

Les opérations RESTCONF sont transposées en modèle de contrôle d'accès sur la base de la méthode HTTP et de la classe de ressources utilisée dans l'opération. Par exemple, une méthode POST sur une ressource de données est considérée comme un accès de "nœud de données en écriture", mais une méthode POST sur une ressource d'opération est considérée comme un accès "operation".

La nouvelle boîte "pre-read data node acc. ctl" dans le diagramme ci-dessous se réfère à l'accès de groupe en lecture car il se rapporte aux ancêtres du nœud de données d'une action ou notification. Par exemple, si une action est définie comme /interfaces/interface/reset-interface, le groupe doit être autorisé (1) à lire /interfaces et /interfaces/interface et (2) exécuter sur /interfaces/interface/reset-interface.



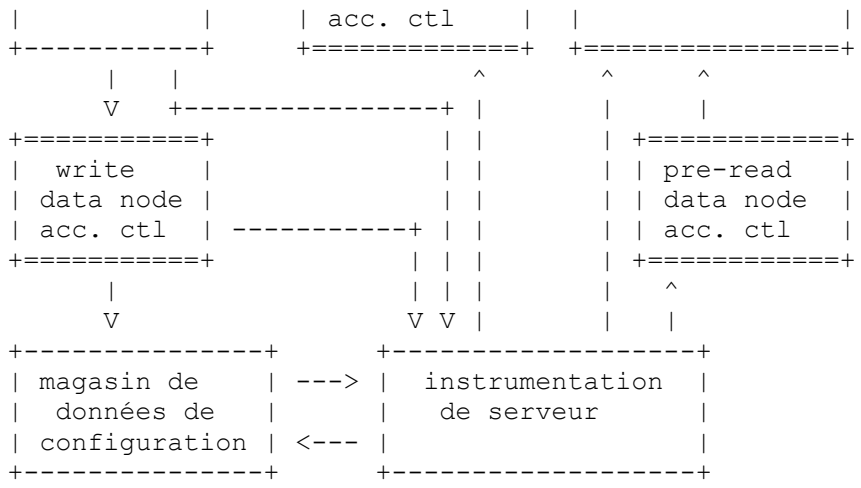


Figure 2

La séquence très générale d'étapes conceptuelles de traitement suivante est exécutée pour chaque message <rpc> reçu, si l'application du contrôle d'accès est activée :

- Pour chaque session active, le contrôle d'accès est appliqué individuellement à tous les messages <rpc> (sauf <close-session>) reçus par le serveur, sauf si la session est identifiée comme session de récupération.

- Si l'opération <action> définie dans la [RFC7950] est invoquée, l'accès en écriture est alors exigé pour toutes les instances de la hiérarchie de nœuds de données qui identifie l'action spécifique dans le magasin de données, et l'accès execute est exigé pour le nœud d'action. Si l'utilisateur n'est pas autorisé à lire tous les nœuds de données spécifiés et à exécuter l'action, la demande est alors rejetée avec une erreur "access-denied" *(accès refusé)*.

- Autrement, si l'utilisateur n'est pas autorisé à exécuter l'opération de protocole spécifiée, la demande est alors rejetée avec l'erreur "accès refusé".

- Si un magasin de données est accédé par l'opération de protocole, le serveur vérifie alors si le client est autorisé à accéder aux nœuds dans le magasin de données. Si l'utilisateur n'est pas autorisé à effectuer l'opération d'accès demandée sur les données demandées, la demande est alors rejetée avec une erreur "accès refusé".

La séquence suivante d'étapes de traitement conceptuelles est exécutée pour chaque événement de notification généré, si l'application du contrôle d'accès est activée :

- L'instrumentation de serveur génère une notification pour un abonnement particulier.

- Si la déclaration "notification" est spécifiée au sein d'une sous arborescence de données, comme spécifié dans la [RFC7950], l'accès en lecture est alors requis pour toutes les instances de la hiérarchie de nœuds de données qui identifie la notification spécifique dans le magasin de données, et l'accès en lecture est requis pour le nœud de notification. Si l'utilisateur n'est pas autorisé à lire tous les nœuds de données spécifiés et le nœud de notification, la notification est alors abandonnée pour cet abonnement.

- Si la déclaration "notification" est une déclaration de niveau supérieur, l'applicateur de contrôle d'accès de notification vérifie le type d'événement de notification, et si c'est un de ceux que l'utilisateur n'est pas autorisé à lire, la notification est alors abandonnée pour cet abonnement.

3.2 Accès à la mémorisation des données

Les mêmes règles de contrôle d'accès s'appliquent à tous les magasins de données qui prennent en charge NACM -- par exemple, le magasin de données de configuration candidat ou le magasin de données de configuration en cours.

Tous les magasins de données de configuration conventionnels et magasins de données d'état de fonctionnement sont contrôlés par le NACM. Les fichiers locaux, les fichiers distants, ou les magasins de données accédés via le paramètre <url> ne sont pas contrôlés par le NACM.

3.2.1 Transposition de nouvelles mémorisations de données en NACM

Il est possible que de nouveaux magasins de données soient définis à l'avenir pour être utilisés avec NETCONF. Le NACM PEUT être appliqué à d'autres magasins de données qui ont des droits d'accès similaires comme défini dans le NACM. Pour appliquer le NACM à un nouveau magasin de données, la spécification du nouveau magasin de données doit définir comment il se transpose en les droits d'accès NACM CRUDX (Create, Read, Update, Delete, eXec). Il est possible que seul un sous ensemble des droits d'accès du NACM soit applicable. Par exemple, seulement un contrôle d'accès de restitution serait nécessaire pour un magasin de données en lecture seule. Les opérations et les droits d'accès qui ne sont pas pris en charge par le modèle NACM CRUDX sortent du domaine d'application du présent document. Un magasin de données n'a pas besoin d'utiliser le NACM, par exemple, la spécification de magasin de données définit quelque chose d'autre ou n'utilise pas le contrôle d'accès.

3.2.2 Droits d'accès

Un petit ensemble de droits d'accès de magasin de données incorporés est nécessaire pour contrôler l'accès à toutes les opérations de protocole possibles, incluant les extensions de fabricant à l'ensemble standard d'opération de protocole.
Le modèle CRUDX peut prendre en charge toutes les opérations de protocole :

- Create : permet au client d'ajouter une nouvelle instance de nœud de données à un magasin de données.
- Read : permet au client de lire une instance de nœud de données d'un magasin de données ou de recevoir des types d'événement de notification.
- Update : permet au client de mettre à jour une instance existante de nœud de données dans un magasin de données.
- Delete : permet au client de supprimer une instance de nœud de données d'un magasin de données.
- eXec : permet au client d'exécuter l'opération.

3.2.3 Méthodes RESTCONF

Le protocole RESTCONF utilise les méthodes HTTP pour effectuer les opérations de magasin de données, similaires à NETCONF. Les procédures NACM étaient à l'origine écrites pour les opérations de protocole NETCONF, de sorte que les méthodes RESTCONF sont transposées en opérations NETCONF pour les besoins du traitement du contrôle d'accès. Les procédures d'application décrites dans ce document s'appliquent aux deux protocoles sauf mention contraire explicite.

L'URI de demande doit être examiné lors du traitement des demandes RESTCONF sur des ressources de données :

- Pour les demandes HEAD et GET, tout nœud de données qui est un nœud ancêtre de la ressource cible est considéré faire partie de la demande de restitution pour les besoins du contrôle d'accès.
- Pour les demandes PUT, PATCH, et DELETE, tout nœud de données qui est un nœud ancêtre de la ressource cible n'est pas considéré faire partie de la demande d'édition pour les besoins du contrôle d'accès. L'opération d'accès pour ces nœuds est considérée être "aucune". L'édition commence à la ressource cible.
- Pour les demandes POST sur des ressources de données, tout nœud de données qui est spécifié dans l'URI de demande, incluant la ressource cible, n'est pas considéré comme faisant partie de la demande d'édition pour les besoins du contrôle d'accès. L'opération d'accès pour ces nœuds est considérée comme étant "aucune". L'édition commence à un nœud fils de la ressource cible, spécifiée dans le corps du message.

Toutes les méthodes RESTCONF ne sont pas soumises au contrôle d'accès. Le tableau suivant spécifie comment chaque méthode est transposée en opération de protocole NETCONF. La valeur "aucune" indique que le NACM n'est pas appliqué à toutes les méthodes spécifiques de RESTCONF.

Méthode	Classe de ressource	Opérations NETCONF	Opération d'accès
OPTIONS	toutes	aucune	aucune
HEAD	toutes	<get>, <get-config>	lire
GET	toutes	<get>, <get-config>	lire
POST	magasin de données, données	<edit-config>	créer
POST	opération	opération spécifiée	exécuter
PUT	données	<edit-config>	créer, mettre à jour
PUT	magasin de données	<copy-config>	mettre à jour
PATCH	données, magasin de données	<edit-config>	mettre à jour
DELETE	données	<edit-config>	supprimer

Tableau 1 : Transposition des méthodes RESTCONF en NETCONF

3.2.4 Opérations <get> et <get-config>

Les droits d'accès NACM ne sont pas directement couplés aux opérations de protocole <get> et <get-config> mais s'appliquent à toutes les opérations <rpc> qui vont résulter en une opération d'accès "read" au magasin de données cible. Ce

paragraphe décrit comment ces droits d'accès s'appliquent aux opérations d'accès spécifiques prises en charge par les opérations de protocole <get> et <get-config>.

Les nœuds de données auxquels le client n'a pas d'accès en lecture sont omis en silence, ainsi que tous descendants, du message <rpc-reply>. Cela est fait pour permettre aux filtres NETCONF pour <get> et <get-config> de fonctionner correctement, au lieu de causer une erreur "accès refusé" parce que les critères de filtre inclurait autrement l'accès en lecture non autorisé pour certains nœuds de données. Pour les besoins du filtrage NETCONF, les critères de choix sont appliqués au sous ensemble de nœuds que l'utilisateur est autorisé à lire, et non à tout le magasin de données.

3.2.5 Opération <edit-config>

Les droits d'accès NACM ne sont pas directement couplés à l'attribut "operation" <edit-config>, bien qu'ils soient similaires. À la place, un droit d'accès NACM s'applique à toutes les opérations de protocole qui vont résulter en une opération d'accès particulière au magasin de données cible. Ce paragraphe décrit comment ces droits d'accès s'appliquent aux opérations d'accès spécifiques prises en charge par l'opération de protocole <edit-config>.

Si l'opération d'accès effective est "none" (c'est-à-dire, default-operation="none") pour un nœud de données particulier, aucun contrôle d'accès n'est alors appliqué à ce nœud de données. Cela est exigé pour permettre l'accès à une sous arborescence au sein d'une plus grande structure de données. Par exemple, un utilisateur peut être autorisé à créer une nouvelle entrée de liste "/interfaces/interface" mais n'être pas autorisé à créer ou supprimer son conteneur parent ("/interfaces"). Si le conteneur "/interfaces" existe déjà dans le magasin de données cible, l'opération effective sera alors "none" pour le nœud "/interfaces" si une entrée de liste "/interfaces/interface" est éditée.

Si l'opération de protocole résulte en la création d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "create" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refuséaccess-denied".

Si l'opération de protocole résulte en la suppression d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "delete" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refusé".

Si l'opération de protocole résulte en la modification d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "update" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refusé".

Une opération <edit-config> "merge" ou "replace" peut inclure des nœuds de données qui n'altèrent pas de portion du magasin de données existant. Par exemple, un nœud conteneur ou liste peut être présent pour des besoins de dénomination mais ne pas réellement modifier le nœud magasin de données correspondant. Ces nœuds de données non altérés sont ignorés par le serveur et n'exigent aucun droit d'accès par le client.

Une opération <edit-config> "merge" peut inclure des nœuds de données mais ne pas inclure de nœud de données fils particulier présent dans le magasin de données. Ces nœuds de données manquants au sein de la portée d'une opération <edit-config> "merge" sont ignorés par le serveur et n'exigent aucun droit d'accès par le client.

Le contenu de nœuds magasin de données spécifiques interdits NE DOIT PAS être exposé dans un élément <rpc-error> dans la réponse.

Une opération <edit-config> peut être cause qu'un nœud de données soit implicitement créé ou supprimé comme effet collatéral implicite d'une opération demandée. Par exemple, une expression YANG when-stmt peut s'évaluer à un résultat différent, causant la suppression, ou la création avec des valeurs par défaut, de nœuds de données ; ou si un nœud de données est créé sous une branche d'un choice-stmt YANG, alors tous les nœuds de données sous les autres branches sont implicitement supprimés. Aucun droit d'accès NACM n'est requis sur des nœuds de données qui sont implicitement changés par suite d'un effet collatéral d'une autre opération permise.

3.2.6 Opération <copy-config>

Le contrôle d'accès pour l'opération <copy-config> exige des attentions particulières parce que l'administrateur peut être en train de remplacer le magasin de données cible tout entier.

Si la source de l'opération de protocole <copy-config> est le magasin de données de configuration en cours et si la cible est le magasin de données de configuration de démarrage, le client doit seulement avoir la permission d'exécuter l'opération de protocole <copy-config>.

Autrement :

- Si la source de l'opération <copy-config> est un magasin de données, les nœuds de données auxquels le client n'a pas

l'accès en lecture sont alors omis en silence.

- Si la cible de l'opération <copy-config> est un magasin de données, le client a besoin de l'accès aux nœuds modifiés. Précisément :
  * Si l'opération de protocole résulte en la création d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "create" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refusé".
  * Si l'opération de protocole résulte en la suppression d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "delete" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refusé".
  * Si l'opération de protocole résulte en la modification d'un nœud magasin de données et si l'utilisateur n'a pas la permission d'accès "update" pour ce nœud, l'opération de protocole est rejetée avec une erreur "accès refusé".

3.2.7 Opération <delete-config>

L'accès à l'opération de protocole <delete-config> est refusé par défaut. La feuille "exec-default" ne s'applique pas à cette opération de protocole. Les règles de contrôle d'accès doivent être explicitement configurées pour permettre l'invocation par une session qui n'est pas de récupération.

3.2.8 Opération <commit>

Le serveur DOIT déterminer les nœuds exacts dans le magasin de données de configuration en cours qui sont réellement différents et seulement vérifier les permissions d'accès "create", "update", et "delete" pour cet ensemble de nœuds, qui peut être vide.

Par exemple, si une session peut lire le magasin de données entier mais seulement changer une feuille, cette session a besoin d'être capable d'éditer et engager cette seule feuille.

3.2.9 Opération <discard-changes>

Il est seulement demandé au client d'avoir la permission d'exécuter l'opération de protocole <discard-changes>. Aucune permission de magasin de données n'est nécessaire.

3.2.10 Opération <kill-session>

L'opération <kill-session> n'altère pas directement un magasin de données. Cependant, elle permet qu'une session interrompe une autre session qui est en train d'éditer un magasin de données.

L'accès à l'opération de protocole <kill-session> est refusé par défaut. La feuille "exec-default" ne s'applique pas à cette opération de protocole. Les règles de contrôle d'accès doivent être explicitement configurées pour permettre l'invocation par une session qui n'est pas de récupération.

3.3 Composants du modèle

Ce paragraphe définit les composants conceptuels relatifs au modèle de contrôle d'accès.

3.3.1 Utilisateurs

Un "utilisateur" est l'entité conceptuelle qui est associée aux permissions d'accès accordées à une certaine session. Un utilisateur est identifié par une chaîne qui est unique au sein du serveur.

Comme décrit dans la [RFC6241], la chaîne "username" est déduite de la couche transport durant l'établissement de session. Si la couche transport ne peut pas authentifier l'utilisateur, la session se termine.

3.3.2 Groupes

L'accès à une opération de protocole NETCONF spécifique est accordé à une session. La session est associée à un groupe (c'est-à-dire, pas à un utilisateur).

Un groupe est identifié par son nom. Tous les noms de groupes sont uniques au sein du serveur.

Le contrôle d'accès est appliqué au niveau des groupes. Un groupe contient zéro, un ou plusieurs membres.

Un membre de groupe est identifié par une chaîne de nom d'utilisateur.

Le même utilisateur peut être un membre de plusieurs groupes.

3.3.3 Session de récupération d'urgence

Le serveur PEUT prendre en charge un mécanisme de récupération de session, qui va outrepasser toutes les applications de contrôle d'accès. Ceci est utile pour restreindre l'accès initial et réparer une configuration de contrôle d'accès endommagée.

3.3.4 Contrôles d'application globale

Il y a cinq contrôles globaux pour contrôler l'application du contrôle d'accès.

3.3.4.1 Commutateur enable-nacm

Un commutateur global "enable-nacm" marche/arrêt est fourni pour activer ou désactiver toute l'application de contrôle d'accès. Lorsque ce commutateur global est réglé à "vrai", toutes les demandes sont confrontées aux règles de contrôle d'accès et ne sont permises que si elles sont configurées à permettre la demande d'accès spécifique. Lorsque ce commutateur global est réglé à "faux", toutes les demandes d'accès sont permises.

3.3.4.2 Commutateur read-default

Un commutateur marche/arrêt "read-default" est fourni pour activer ou désactiver la réception de données par l'accès par défaut dans les réponses et les notifications. Lorsque le commutateur global "enable-nacm" est réglé à "vrai", ce commutateur global est pertinent si aucune règle de contrôle d'accès correspondante ne se trouve permettre ou interdire explicitement l'accès en lecture aux données du magasin de données demandé ou au type d'événement de notification.

Lorsque ce commutateur global est réglé à "permit" et qu'aucune règle de contrôle d'accès correspondant n'est trouvée pour l'événement de lecture ou de notification du magasin de données demandé, l'accès est permis.

Lorsque ce commutateur global est réglé à "deny" et qu'aucune règle de contrôle d'accès correspondante n'est trouvée pour l'événement de lecture ou notification du magasin de données demandé, l'accès est refusé. Cela signifie que les données demandées ne sont pas envoyées au client. Voir les détails à l'étape 11 du paragraphe 3.4.5.

3.3.4.3 Commutateur write-default

Un commutateur marche/arrêt "write-default" est fourni pour permettre ou interdire à l'accès par défaut d'altérer les données de configuration. Lorsque le commutateur global "enable-nacm" est réglé à "vrai", ce commutateur global est pertinent si aucune règle de contrôle d'accès correspondante n'est trouvée pour explicitement permettre ou refuser l'accès en écriture aux données du magasin de données demandées.

Lorsque ce commutateur global est réglé à "permit" et qu'aucune règle de contrôle d'accès correspondante n'est trouvée pour l'écriture du magasin de données demandée, l'accès est permis.

Lorsque ce commutateur global est réglé à "deny" et qu'aucune règle de contrôle d'accès correspondante n'est trouvée pour l'écriture du magasin de données demandée, l'accès est refusé. Voir les détails à l'étape 12 du paragraphe 3.4.5.

3.3.4.4 Commutateur exec-default

Un commutateur marche/arrêt "exec-default" est fourni pour permettre ou interdire à l'accès par défaut d'exécuter les opérations de protocole. Lorsque le commutateur global "enable-nacm" est réglé à "vrai", ce commutateur global est pertinent si aucune règle de contrôle d'accès correspondante n'est trouvée pour explicitement permettre ou refuser l'accès à l'opération de protocole NETCONF demandée.

Lorsque ce commutateur global est réglé à "permit" et qu'aucune règle de contrôle d'accès correspondante n'est trouvée pour l'opération de protocole NETCONF demandée, l'accès est permis.

Lorsque ce commutateur global est réglé à "deny" et qu'aucune règle de contrôle d'accès correspondante n'est trouvée pour l'opération de protocole NETCONF demandée, l'accès est refusé. Voir les détails à l'étape 12 du paragraphe 3.4.4 et à l'étape 13 du paragraphe 3.4.5.

3.3.4.5 Commutateur enable-external-groups

Lorsque ce commutateur global est réglé à "vrai", les noms de groupes rapportés par la couche transport pour une session sont utilisés avec les noms de groupes configurés en local pour déterminer les règles de contrôle d'accès pour la session.

Lorsque ce commutateur est réglé à "faux", les noms de groupe rapportés par la couche transport sont ignorés par le NACM.

3.3.5 Règles de contrôle d'accès

Quatre types de règles sont disponibles dans le NACM :

règle de module : contrôle l'accès pour les définitions dans un module YANG spécifique, identifié par son nom.
règle d'opération de protocole : contrôle l'accès pour une opération de protocole spécifique, identifiée par son module YANG et son nom.

règle de nœud de données : contrôle l'accès pour un nœud de données spécifique et ses descendants, identifié par sa localisation de chemin au sein du document XML conceptuel pour le nœud de données.

règle de notification : contrôle l'accès pour un type d'événement de notification spécifique, identifié par son module YANG et son nom.

3.4 Procédures d'application de contrôle d'accès

Il y a six phases distinctes qui doivent être effectuées, dont quatre sont relatives au modèle de traitement de message NETCONF (paragraphe 3.1.3):
1. fonctionnement initial
2. établissement de session
3. traitement de l'erreur "accès refusé"
4. validation de message RPC entrant
5. validation d'accès au nœud de données
6. autorisation <notification> sortante

De plus, le mode de démarrage initial pour un serveur NETCONF, l'établissement de session, et les procédures de traitement de l'erreur "accès refusé" doivent aussi être pris en compte.

Le serveur DOIT utiliser les règles de contrôle d'accès en vigueur au moment où il commence le traitement du message. Les mêmes règles de contrôle d'accès DOIVENT rester en vigueur pendant le traitement du message entier.

3.4.1 Fonctionnement initial

Au tout début du démarrage du serveur NETCONF, la configuration du contrôle d'accès ne sera probablement pas présente. Si elle n'est pas là, un serveur NE DOIT PAS permettre d'accès en écriture à un rôle de session, sauf si c'est une session de récupération.

Les règles d'accès sont appliquées à chaque fois qu'une demande est initiée par une session d'utilisateur. Le contrôle d'accès n'est pas appliqué pour les demandes d'accès initiées par le serveur, comme le chargement initial du magasin de données de configuration en cours durant l'amorçage.

3.4.2 Établissement de session

Le modèle de contrôle d'accès s'applique spécifiquement au contenu XML bien formé transféré entre un client et un serveur après l'achèvement de l'établissement de session et après que l'échange de <hello> se termine avec succès.

Une fois l'établissement de session achevé et l'utilisateur authentifié, la couche transport rapporte au serveur NETCONF le nom de l'utilisateur et un ensemble éventuellement vide de noms de groupe associé à l'utilisateur. Le serveur NETCONF va appliquer les règles de contrôle d'accès, sur la base du nom d'utilisateur fourni, des noms de groupe, et des données de configuration mémorisées au serveur.

3.4.3 Traitement de l'erreur "accès refusé"

L'étiquette d'erreur "accès refusé" est générée lorsque le système de contrôle d'accès refuse l'accès à une demande

d'invoquer une opération de protocole ou une demande d'effectuer une opération particulière d'accès sur le magasin de données de configuration.

Un serveur NE DOIT PAS inclure des informations que le client n'a pas la permission de lire dans un élément <error-info> au sein de la réponse <rpc-error>.

3.4.4 Validation de message RPC entrant

Le diagramme ci-dessous montre la structure conceptuelle de base du modèle de traitement de contrôle d'accès pour les messages NETCONF <rpc> entrants au sein d'un serveur.

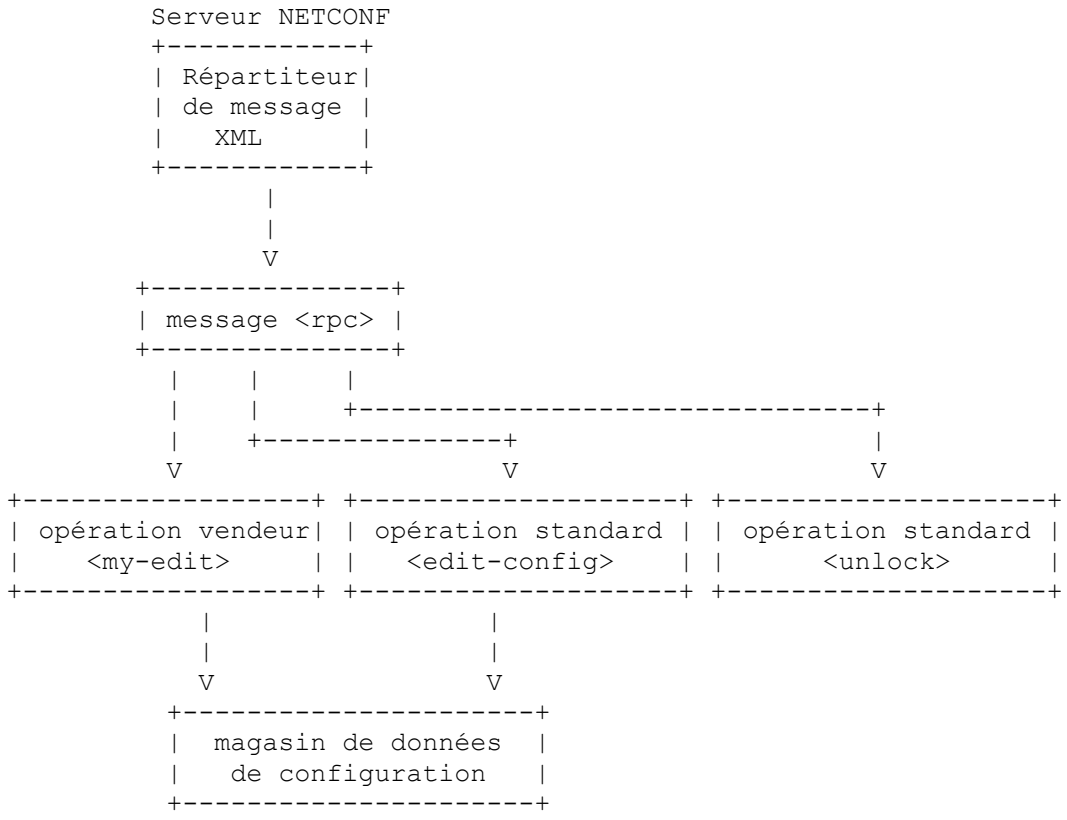


Figure 3

Le contrôle d'accès commence avec le répartiteur de message.

Après que le serveur a validé l'élément <rpc> et déterminé l'URI d'espace de noms et le nom d'élément de l'opération de protocole demandée, le serveur vérifie que l'utilisateur est autorisé à invoquer l'opération de protocole.

Le serveur DOIT autoriser séparément chaque opération de protocole en suivant ces étapes :

1. Si la feuille "enable-nacm" est réglée à "faux", l'opération de protocole est alors permise.
2. Si la session demandeuse est identifiée comme session de récupération, l'opération de protocole est alors permise.
3. Si l'opération demandée est l'opération de protocole <close-session> NETCONF, l'opération de protocole est permise.
4. Vérifier toutes les entrées "group" pour voir si une d'elles contient une entrée "nom d'utilisateur" qui est égale au nom d'utilisateur pour la session qui fait la demande. Si la feuille "enable-external-groups" est "vrai", ajouter ces groupes à l'ensemble des groupes fournis par la couche transport.
5. Si aucun groupe n'est trouvé, continuer à l'étape 10.
6. Traiter toutes les entrées de liste de règles, dans l'ordre où elle apparaissent dans la configuration. Si une liste feuille "group" d'une liste de règles ne correspond à aucun des groupes de l'utilisateur, passer à la prochaine entrée de la liste de règles.
7. Pour chaque entrée de liste de règles trouvée, traiter toutes les règles, dans l'ordre, jusqu'à trouver une règle qui corresponde à l'opération d'accès demandée. Une règle correspond si tous les critères suivants sont satisfaits :
   * La feuille "module-name" de la règle est "*" ou égale au nom du module YANG où l'opération de protocole est définie.
   * Soit (1) la règle n'a pas de "rule-type" défini, soit (2) le "rule-type" est "protocol-operation" et le "rpc-name" est "*" ou égal au nom de l'opération de protocole demandée.

* La feuille "access-operations" de la règle a le bit "exec" établi ou a la valeur spéciale "*".

8. Si une règle correspondante est trouvée, la feuille "action" est alors vérifiée. Si elle est égale à "permit", alors l'opération de protocole est permise; sinon, elle est refusée.
9. À ce point, aucune règle correspondante n'a été trouvée dans une entrée de liste de règle.
10. Si l'opération de protocole demandée est définie dans un module YANG annoncé dans les capacités de serveur et si la déclaration "rpc" contient une déclaration "nacm:default-deny-all", l'opération de protocole est alors refusée.
11. Si l'opération de protocole demandée est le <kill-session> ou <delete-config> NETCONF, alors l'opération de protocole est refusée.
12. Si la feuille "exec-default" est réglée à "permit", permettre alors l'opération de protocole ; sinon, refuser la demande.

Si l'utilisateur n'est pas autorisé à invoquer l'opération de protocole, alors une <rpc-error> est générée avec les informations suivantes :

error-tag : accès refusé

error-path : Identifie l'opération de protocole demandée. L'exemple suivant représente l'opération de protocole <edit-config> dans l'espace de noms NETCONF de base :

```
<error-path
      xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
        /nc:rpc/nc:edit-config
</error-path>
```

Si un magasin de données est accédé, soit directement, soit comme effet collatéral de l'opération de protocole, le serveur DOIT alors intercepter l'opération d'accès et s'assurer que l'utilisateur est autorisé à effectuer l'opération d'accès demandée sur les données spécifiées, comme défini au paragraphe 3.4.5.

3.4.5 Validation d'accès du nœud de données

Si (1) un nœud de données au sein d'un magasin de données est accédé ou (2) si une action ou notification est liée à un nœud de données, le serveur DOIT alors s'assurer que l'utilisateur est autorisé à effectuer l'opération d'accès "read", "create", "update", "delete", ou "execute" demandée sur le nœud de données spécifié.

Si l'exécution d'une action est demandée, le serveur DOIT s'assurer que l'utilisateur est autorisé à effectuer l'opération d'accès "execute" sur l'action demandée.

Si une notification liée à un nœud de données est générée, le serveur DOIT s'assurer que l'utilisateur est autorisé à effectuer l'opération d'accès "read" sur la notification demandée.

La demande d'accès au nœud de données est autorisée en suivant ces étapes :

1. Si la feuille "enable-nacm" est réglée à "faux", l'opération d'accès est alors permise.
2. Si la session demandeuse est identifiée comme session de récupération, l'opération d'accès est alors permise.
3. Vérifier toutes les entrées "group" pour voir si l'une d'elles contient une entrée "user-name" égale au nom de l'utilisateur pour la session qui fait la demande. Si la feuille "enable-external-groups" est "vrai", ajouter ces groupes à l'ensemble des groupes fournis par la couche transport.
4. Si aucun groupe n'est trouvé, continuer à l'étape 9.
5. Traiter toutes les entrées de liste de règles, dans l'ordre où elle apparaissent dans la configuration. Si une liste de feuille "group" d'une liste de règles ne correspond à aucun des groupes de l'utilisateur, passer à la prochaine entrée de liste de règles.
6. Pour chaque entrée de liste de règles trouvée, traiter toutes les règles, dans l'ordre, jusqu'à ce qu'une règle qui corresponde à l'opération d'accès demandée soit trouvée. Une règle correspond si tous les critères suivants sont satisfaits :
   * La feuille "module-name" de la règle est "*" ou égale au nom du module YANG où le nœud de données demandé est défini.
   * Soit (1) la règle n'a pas de "rule-type" défini, soit (2) le "rule-type" est "data-node" et le "path" correspond au nœud de données, au nœud d'action, ou au nœud de notification demandé. Un chemin est considéré correspondre si le nœud demandé est celui qui est spécifié par le chemin ou est un nœud descendant sur le chemin.
   * Pour une opération d'accès "read", la feuille "access-operation" de la règle a le bit "read" établi ou a la valeur spéciale "*".
   * Pour une opération d'accès "create", la feuille "access-operation" de la règle a le bit "create" établi ou a la valeur spéciale "*".
   * Pour une opération d'accès "delete", la feuille "access-operation" de la règle a le bit "delete" établi ou a la valeur

spéciale "*".
* Pour une opération d'accès "update", la feuille "access-operation" de la règle a le bit "update" établi ou a la valeur spéciale "*".
* Pour une opération d'accès "execute", la feuille "access-operation" de la règle a le bit "exec" établi ou a la valeur spéciale "*".

7. Si une règle correspondante est trouvée, la feuille "action" est alors vérifiée. Si elle est égale à "permit", l'accès au nœud de données est alors permis; sinon, il est refusé. Pour une opération d'accès "read", "refusé" signifie que les données demandées ne sont pas retournées dans la réponse.
8. À ce point, aucune règle correspondante n'a été trouvée dans une entrée de liste de règles.
9. Pour une opération d'accès "read", si le nœud de données demandé est défini dans un module YANG annoncé dans les capacités du serveur et si la déclaration de définition des données contient une déclaration "nacm:default-deny-all", le nœud de données demandé et tous ses descendants ne sont pas inclus dans la réponse.
10. Pour une opération d'accès "write", si le nœud de données demandé est défini dans un module YANG annoncé dans les capacités du serveur et si la déclaration de définition des données contient une déclaration "nacm:default-deny-write" ou "nacm:default-deny-all", la demande d'accès est alors refusée pour le nœud de données et tous ses descendants.
11. Pour une opération d'accès "read", si la feuille "read-default" est réglée à "permit", on inclut alors le nœud de données demandé dans la réponse ; autrement, on n'inclut pas le nœud de données demandé ni aucun de ses descendants dans la réponse.
12. Pour une opération d'accès "write", si la feuille "write-default" est réglée à "permit", on permet alors la demande d'accès au nœud de données ; sinon, on refuse la demande.
13. Pour une opération d'accès "execute", si la feuille "exec-default" est réglée à "permit", on permet alors la demande ; sinon, on la refuse.

3.4.6 Autorisation <notification> sortante

La configuration des règles de contrôle d'accès spécifiquement pour les nœuds descendants du type d'événement de notification sort du domaine d'application du présent document. Si l'utilisateur est autorisé à recevoir le type d'événement notification, il est alors aussi autorisé à recevoir toutes les données qu'elle contient.

Si la notification est spécifiée au sein d'une sous arborescence de données, comme spécifié dans la [RFC7950], l'accès en lecture à la notification est alors exigé. Le traitement se poursuit comme décrit au paragraphe 3.4.5.

La figure suivante montre le modèle conceptuel de traitement de message pour les messages <notification> sortants.

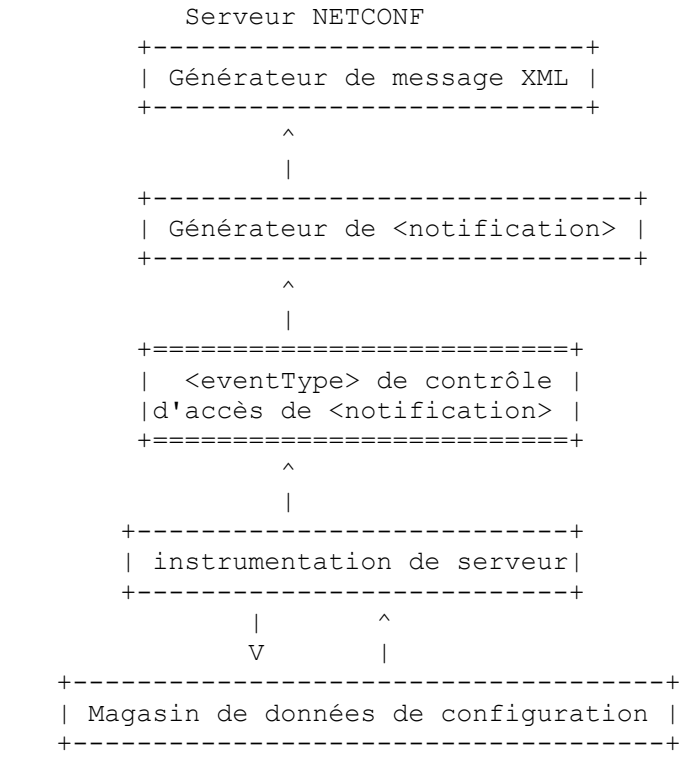


Figure 4

La génération d'une notification pour un abonnement spécifique [RFC5277] est autorisée en suivant ces étapes :
1. Si la feuille "enable-nacm" est réglée à "faux", la notification est alors permise.

2. Si la session est identifiée comme session de récupération, la notification est alors permise.
3. Si la notification est le type d'événement NETCONF <replayComplete> ou <notificationComplete> [RFC5277], la notification est alors permise.
4. Vérifier toutes les entrées "group" pour voir si une d'elles contient une entrée "user-name" égale au nom d'utilisateur pour la session qui fait la demande. Si la feuille "enable-external-groups" est "vrai", ajouter ces groupes à l'ensemble des groupes fourni par la couche transport.
5. Si aucun groupe n'est trouvé, continuer à l'étape 10.
6. Traiter toutes les entrées de liste de règles, dans l'ordre où elles apparaissent dans la configuration. Si une liste de feuilles "group" de liste de règle ne correspond à aucun des groupes de l'utilisateur, passer à la prochaine entrée de la liste de règles.
7. Pour chaque entrée de liste de règles trouvée, traiter toutes les règles, dans l'ordre, jusqu'à ce qu'une règle qui corresponde à l'opération d'accès demandée soit trouvée. Une règle correspond si tous les critères suivants sont satisfaits :
   * la feuille "module-name" de la règle est "*" ou égale au nom du module YANG où la notification est définie ;
   * soit (1) la règle n'a pas de "rule-type" défini, soit (2) le "rule-type" est "notification" et le "notification-name" est "*" ou égal au nom de la notification ;
   * la feuille "access-operations" de la règle a le bit "read" établi ou a la valeur spéciale "*".
8. Si une règle correspondante est trouvée, la feuille "action" est alors vérifiée. Si elle est égale à "permit", on permet alors la notification ; autrement, on abandonne la notification pour l'abonnement associé.
9. Autrement, aucune règle correspondante n'a été trouvée dans une entrée de liste de règles.
10. Si la notification demandée est définie dans un module YANG annoncé dans les capacités de serveur et si la déclaration "notification" contient une déclaration "nacm:default-deny-all", la notification est alors abandonnée pour l'abonnement associé.
11. Si la feuille "read-default" est réglée à "permit", on permet alors la notification ; autrement, on abandonne la notification pour l'abonnement associé.

3.5 Définitions du modèle de données

3.5.1 Organisation des données

Le diagramme suivant met en lumière le contenu et la structure du module NACM YANG.

```
module : ietf-netconf-acm
   +--rw nacm
     +--rw enable-nacm?                 booléen
     +--rw read-default?                type action
     +--rw write-default?               type action
     +--rw exec-default?                type action
     +--rw enable-external-groups?      booléen
     +--ro denied-operations            yang:zero-based-counter32
     +--ro denied-data-writes           yang:zero-based-counter32
     +--ro denied-notifications         yang:zero-based-counter32
     +--rw groups
     | +--rw group*                     [nom]
     |    +--rw name                    type nom de groupe
     |    +--rw user-name*              type nom d'utilisateur
     +--rw rule-list*                   [nom]
       +--rw name                       chaîne
       +--rw group*                     union
       +--rw rule*                      [nom]
         +--rw name                     chaîne
         +--rw module-name?             union
         +--rw (rule-type)?
         | +--:(protocol-operation)
         | | +--rw rpc-name?            union
         | +--:(notification)
         | | +--rw notification-name?   union
         | +--:(data-node)
         |    +--rw path                identifiant d'instance de nœud
         +--rw access-operations?       union
         +--rw action                   type action
         +--rw comment?                 chaîne
```

3.5.2 Module YANG

Le module YANG suivant spécifie le contenu normatif NETCONF qui DOIT être pris en charge par le serveur.

Le module YANG "ietf-netconf-acm" importe les définitions de type de la [RFC6991].

<DÉBUT DU CODE> fichier "ietf-netconf-acm@2018-02-14.yang" module ietf-netconf-acm {

namespace "urn:ietf:params:xml:ns:yang:ietf-netconf-acm";

prefix nacm;

import ietf-yang-types { prefix yang; }

organisation "Groupe de travail IETF NETCONF (Network Configuration)" ;

contact :
"WG Web: <https://datatracker.ietf.org/wg/netconf/>
WG List: <mailto:netconf@ietf.org>

Auteur : Andy Bierman <mailto:andy@yumaworks.com>
Auteur : Martin Bjorklund <mailto:mbj@tail-f.com>";

Description : "Modèle de contrôle d'accès de configuration réseau".
Copyright (c) 2012 - 2018 IETF Trust et les personnes identifiées comme auteurs du code. Tous droits réservés. La redistribution et l'utilisation en formes source et binaire, avec ou sans modification, est permise selon, et sous réserve des termes de licence contenus dans la licence BSD simplifiée présentée à la Section 4.c du règlement intérieur de l'IETF Trust relative aux documents de l'IETF (https://trustee.ietf.org/license-info).
Cette version de ce module YANG fait partie de la RFC 8341 ; voir les notices légales dans la RFC elle-même.";

Révision "2018-02-14" {
Description "Ajout de la prise en charge des actions et notifications YANG 1.1 liées aux nœuds de données. Précision de comment les extensions NACM peuvent être utilisées par d'autres modèles de données.";
Référence : "RFC 8341 : Modèle de contrôle d'accès de configuration réseau"; }

Révision "2012-02-22" {
Description : "version. initiale";
référence : "RFC 6536: Network Configuration Protocol (NETCONF) Access Control Model";
}

/* Déclarations d'extension */

extension default-deny-write {description : "Utilisée pour indiquer que le nœud de modèle de données représente un paramètre sensible du système de sécurité. Si elle est présente, le serveur NETCONF va seulement permettre à la "session de récupération" désignée d'avoir l'accès en écriture au nœud. Une règle explicite de contrôle d'accès est requise pour tous les autres utilisateurs. Si le module NACM est utilisé, il doit alors être activé (c'est-à-dire, l'objet /nacm/enable-nacm est "vrai") ou cette extension est ignorée. l'extension "default-deny-write" PEUT apparaître dans une déclaration de définition de données. Elle est ignorée autrement."; }

extension default-deny-all {
description : "Utilisée pour indiquer que le nœud de modèle de données contrôle un paramètre de système de sécurité très sensible. Si elle est présente, le serveur NETCONF va seulement permettre à la "session de récupération" désignée d'avoir l'accès en lecture, écriture, ou exécution au nœud. Une règle explicite de contrôle d'accès est requise pour tous les autres utilisateurs. Si le module NACM est utilisé, il doit alors être activé (c'est-à-dire, l'objet /nacm/enable-nacm est "vrai") ou cette extension est ignorée. L'extension "default-deny-all" PEUT apparaître dans une déclaration de définition de données, une déclaration "rpc", ou une déclaration "notification". Elle est ignorée autrement." }

/* * Types dérivés */

typedef user-name-type {
type chaîne {

```
        longueur "1..max";
      }
      description : "Chaîne de nom d'utilisateur d'utilisation générale.";
    }

typedef matchall-string-type {
      type chaîne {
        pattern '\*';
      }
description : "La chaîne contenant une seule astérisque '*' est utilisée pour représenter conceptuellement toutes les valeurs possibles pour la feuille particulière qui utilise ce type de données.";
    }

typedef access-operations-type {
      type bits {
bit create {
description : "Toute opération de protocole qui crée un nouveau nœud de données.";        }
bit read {
description : "Toute opération de protocole ou notification qui retourne la valeur d'un nœud de données.";
        }
bit update {
description : "Toute opération de protocole qui altère un nœud de données existant.";
        }
bit delete {description : "Toute opération de protocole qui supprime un nœud de données.";
        }
bit exec {
description : "Accès en exécution à l'opération de protocole spécifiée.";
        }
      }
description : "Opération d'accès.";
    }

typedef group-name-type {
      type chaîne {
        longueur "1..max";
        pattern '[^\*].*';
      }
description : "Nom du groupe administratif auquel des utilisateurs peuvent être alloués.";
    }

typedef action-type {
      type enumeration {
enum permit {
description : "L'action demandée est permise.";
        }
enum deny {
description : "L'action demandée est refusée.";
        }
      }
description : "Action prise par le serveur quand une règle particulière correspond.";
    }

typedef node-instance-identifier {
      type yang:xpath1.0;
description : "Expression de chemin utilisée pour représenter un nœud de données, action, ou chaîne d'identifiant d'instance de notification spéciale. Une valeur d'identifiant d'instance de nœud est une expression d'identifiant d'instance YANG sans restriction. Toutes les règles d'un identifiant d'instance s'appliquent, excepté que les prédicats des clés sont facultatifs. Si un prédicat de clé manque, l'identifiant d'instance de nœud représente toutes les instances de serveur possibles pour cette clé. Cette expression Path de langage XML (XPath) est évaluée dans le contexte suivant :
o  L'ensemble des déclarations d'espace de noms dont la portée est l'élément feuille où ce type est utilisé.
o  L'ensemble des liens de variables contient une variable, 'USER', qui contient le nom de l'utilisateur de la session en cours.
o  La fonction "library" est la fonction de bibliothèque centrale, mais à cause des restrictions syntaxiques sur l'identifiant
```

```
    d'instance, aucune fonction n'est permise.
o   Le nœud de contexte est le nœud racine dans l'arborescence des données.
L'arborescence accessible inclut des actions et notifications liées aux nœuds de données.";
    }

/*     * Déclarations de définition de données */

conteneur nacm {
     nacm:default-deny-all;
description : "Paramètres pour le modèle NETCONF de contrôle d'accès." ;
feuille enable-nacm {
       type booléen ;
       défaut "vrai";
description : "Active ou désactive toutes les applications NETCONF de contrôle d'accès. Si c'est "vrai", l'application est activée. Si c'est "faux", l'application est désactivée." ;
     }

feuille read-default {
       type action ;
       défaut "permit";
description : "Contrôle que l'accès en lecture est accordé si aucune règle appropriée n'est trouvée pour une demande de lecture particulière." ;
     }

feuille write-default {
       type action ;
       défaut "deny";
description : "Contrôle que l'accès à la création, mise à jour ou suppression est accordé si aucune règle appropriée n'est trouvée pour une demande d'écriture particulière." ;
     }

feuille exec-default {
       type action ;
       défaut "permit";
description : "Contrôle que l'accès en exécution est accordé si aucune règle appropriée n'est trouvée pour une demande d'opération de protocole particulière." ;
     }

feuille enable-external-groups {
       type booléen ;
       défaut "vrai";
description : "Contrôle que le serveur utilise les groupes rapportés par la couche transport NETCONF lorsque elle affecte l'utilisateur à un ensemble de groupes NACM. Si cette feuille a la valeur "faux", tous les noms de groupe rapportés par la couche transport sont ignorés par le serveur." ;
     }

feuille denied-operations {
       type yang:zero-based-counter32;
       config faux ;
       obligatoirement vrai ;
description : "Nombre de fois depuis le dernier redémarrage du serveur qu'une demande d'opération de protocole a été refusée." ;
     }

feuille denied-data-writes {
       type yang:zero-based-counter32;
       config faux ;
       obligatoirement vrai ;
description : "Nombre de fois depuis le dernier redémarrage du serveur qu'une demande d'opération de protocole d'altérer un magasin de données de configuration a été refusée." ;
     }

feuille denied-notifications {
```

```
        type yang:zero-based-counter32;
        config faux ;
        obligatoirement vrai ;
description : "Nombre de fois depuis le dernier redémarrage du serveur qu'une notification a été abandonnée pour un abonnement parce que l'accès au type d'événement est refusé." ;
      }

conteneur groups {
description : "Groupes de contrôle d'accès NETCONF." ;

liste group {
         key name ;
description : "Entrée de groupe NACM. Cette liste va seulement contenir des entrées configurées, aucune entrée apprise d'un protocole de transport." ;

feuille name {            type group-name-type;
description : "Nom de groupe associé à cette entrée." ;
         }

liste de feuilles user-name {
           type user-name-type;
description : "Chaque entrée identifie le nom d'utilisateur d'un membre du groupe associé à cette entrée." ;
         }
       }
      }

liste rule-list {
        key name ;
        ordonné par utilisateur ;
description : "Collection ordonnée de règles de contrôle d'accès.";

feuille name {
          type chaîne {
            longueur "1..max";
          }
description : "Nom arbitraire alloué à la liste de règles." ;
        }

liste de feuilles group {
          type union {
            type matchall-string-type ;
            type group-name-type ;
          }
description : "Liste des groupes administratifs auxquels seront alloués les droits d'accès associés définis par la liste "rule". La chaîne '*' indique que tous les groupes s'appliquent à l'entrée.";
        }

liste rule {
          key name ;
          ordonnée par utilisateur ;
description : "Une règle de contrôle d'accès. Les règles sont traitées dans l'ordre défini par utilisateur jusqu'à ce qu'une correspondance soit trouvée. Une règle correspond si "module-name", "rule-type", et "access-operations" correspondent à la demande. Si une règle correspond, la feuille "action" détermine si l'accès est accordé ou non." ;

feuille name {
            type chaîne {
              longueur "1..max";
            }
description : "Nom arbitraire alloué à la règle." ;
          }

feuille module-name {
            type union {
```

```
          type matchall-string-type;
          type chaîne ;
        }
        défaut "*";
description : "Nom du module associé à cette règle. Cette feuille correspond si elle a la valeur '*' ou si l'objet accédé est défini dans le module avec le nom de module spécifié." ;
      }

choix rule-type {
description : "Ce choix correspond si toutes les feuilles présentes dans la règle correspondent à la demande. Si aucune feuille n'est présente, le choix correspond à toutes les demandes." ;
        cas protocol-operation {
          feuille rpc-name {
            type union {
              type matchall-string-type;
              type chaîne ;
            }
description : "Cette feuille correspond si elle a la valeur '*' ou si sa valeur est égale au nom de l'opération de protocole demandée." ;
          }
        }
        cas notification {
          feuille notification-name {
            type union {
              type matchall-string-type;
              type chaîne ;
            }
description : "Cette feuille correspond si elle a la valeur '*' ou si sa valeur est égale au nom de la notification demandée." ;
          }
        }
        cas data-node {
          feuille path {
            type node-instance-identifier;
        obligatoirement vrai ;
description : "Identifiant d'instance de nœud de données associé au nœud de données, action, ou notification contrôlé par cette règle. Les identifiants d'instance de données de configuration ou d'état commencent par un nœud de données. de niveau supérieur. Un identifiant d'instance complet est exigé pour ce type de valeur de chemin. La valeur spéciale '/' se réfère à tous les contenus de magasin de données possibles." ;
          }
        }
      }

feuille access-operations {
      type union {
        type matchall-string-type;
        type access-operations-type;
      }
      défaut "*";
description : "Opérations d'accès associées à cette règle. Cette feuille correspond si elle a la valeur '*' ou si le bit correspondant à l'opération demandée est établi.";
    }

feuille action {
      type action ;
      obligatoirement vrai ;
description : "Action de contrôle d'accès associée à la règle. Si une règle a été déterminée correspondre à une demande particulière, cet objet est alors utilisé pour déterminer si la demande sera permise ou refusée." ;
    }

feuille comment {
        type chaîne ;
description : "Description textuelle de la règle d'accès." ;
    }
```

```
      }
    }
  }
}
```

<FIN DU CODE>

4. Considérations relatives à l'IANA

Le présent document réutilise l'URI pour "ietf-netconf-acm" dans le registre "IETF XML".

Le présent document met à jour l'enregistrement du module dans le registre "Noms de module YANG" pour faire référence à la présente RFC au lieu de la RFC 6536 pour "ietf-netconf-acm". Suivant le format indiqué dans la [RFC6020], il a été enregistré ce qui suit :

Nom : ietf-netconf-acm
Espace de noms : urn:ietf:params:xml:ns:yang:ietf-netconf-acm
Préfixe : nacm
Référence : RFC 8341

5. Considérations sur la sécurité

Le module YANG spécifié dans le présent document définit un schéma pour des données qui est conçu pour l'accès via des protocoles de gestion de réseau tels que NETCONF [RFC6241] ou RESTCONF [RFC8040]. La couche NETCONF inférieure est la couche de transport sûr, et le transport sûr de mise en œuvre obligatoire est Secure Shell (SSH) [RFC6242]. La couche RESTCONF inférieure est HTTPS, et le transport de mise en œuvre obligatoire est TLS [RFC5246].

Le modèle NETCONF de contrôle d'accès [RFC8341] donne le moyen de restreindre l'accès pour des utilisateurs NETCONF ou RESTCONF particuliers à un sous ensemble préconfiguré de toutes les opérations NETCONF ou RESTCONF de protocole et de contenu disponibles.

Il y a un risque en rapport avec l'absence d'application de contrôle d'accès pour les options RESTCONF et les méthodes PATCH. Le risque est que la réponse à OPTIONS et PATCH puisse varier sur la base de la présence ou l'absence d'une ressource correspondant au chemin d'URL. Si c'est le cas, cela peut être utilisé très facilement pour sonder la présence ou l'absence de valeurs dans une arborescence. Donc, un serveur NE DOIT PAS faire varier ses réponses sur la base de l'existence de la ressource sous-jacente, ce qui indiquerait la présence ou l'absence des instances de ressources. En particulier, les serveurs ne devraient pas exposer d'informations d'instances avant de s'assurer que le client a les permissions d'accès nécessaires pour obtenir ces informations. Dans ces cas, les serveurs sont supposés toujours retourner la réponse d'erreur "accès refusé".

Un certain nombre de nœuds de données définis dans ce module YANG sont écrivables/créables/supprimables (c'est-à-dire, "config true", qui est par défaut). Ces nœuds de données peuvent être considérés comme sensibles ou vulnérables dans certains environnements de réseau. Les opérations d'écriture (par exemple, "edit-config") sur ces nœuds de données sans protection appropriée peuvent avoir un effet négatif sur le fonctionnement du réseau. La sous arborescence /nacm toute entière se rapporte à la sécurité. Les paragraphes qui suivent développent ce point.

Cette section est consacrée aux problèmes que rencontre un administrateur pour configurer un serveur NETCONF avec le NACM.

5.1 Considérations de configuration et de surveillance du NACM

La configuration du système de contrôle d'accès est très sensible à la sécurité du système. Un serveur peut choisir de ne permettre aucune configuration d'utilisateur pour certaines de ses portions, comme le niveau global de sécurité ou les groupes que permet l'accès aux ressources du système.

Par défaut, l'application du NACM est activée. Par défaut, l'accès en écriture à tous les contenus de magasin de données est activé (sauf si "nacm:default-deny-all" est spécifié pour la définition des données) et l'accès "exec" est activé pour le fonctionnement sûr du protocole. Un administrateur doit s'assurer que le NACM est activé et aussi décider si les paramètres d'accès par défaut sont réglés de façon appropriée. Il s'assure que les nœuds de données suivants sont correctement

configurés :
- o /nacm/enable-nacm ("vrai" par défaut)
- o /nacm/read-default ("permit" par défaut)
- o /nacm/write-default ("deny" par défaut)
- o /nacm/exec-default ("permit" par défaut)

Un administrateur doit interdire l'accès en écriture à tous les objets configurables dans ce modèle de données.

Si l'accès en écriture est permis pour la configuration des règles de contrôle d'accès, il faut alors veiller à ce qu'elles n'interrompent pas l'application du contrôle d'accès. Par exemple, si les règles de contrôle d'accès du NACM sont éditées directement dans le magasin de données de configuration en cours (c'est-à-dire, si la capacité :writable-running est prise en charge et utilisée) il faut alors veiller à ne pas permettre un accès non prévu pendant l'édition.

Un administrateur doit s'assurer que la traduction d'une identité d'utilisateur dépendante du transport ou de la mise en œuvre en nom d'utilisateur NACM est unique et correcte. Cette exigence est spécifiée en détail au paragraphe 2.2 de la [RFC6241].

Un administrateur doit savoir que les structures de données YANG représentant les règles de contrôle d'accès (/nacm/rule-list et /nacm/rule-list/rule) sont ordonnées par le client. Le serveur va évaluer les règles de contrôle d'accès conformément à leur ordre conceptuel relatif au sein du magasin de données de configuration en cours.

Noter que la structure de données /nacm/groups contient les noms de groupe administratifs utilisés par le serveur. Ces noms de groupe peuvent être configurés en local et/ou fournis par un protocole externe, comme RADIUS [RFC2865] [RFC5607].

Un administrateur doit savoir les propriétés de sécurité de tout protocole externe utilisé par la couche transport pour déterminer les noms de groupe. Par exemple, si ce protocole ne protège pas contre les attaques par interposition, un attaquant peut être capable d'injecter des noms de groupe qui sont configurés dans le NACM de sorte qu'un utilisateur obtienne plus de permissions qu'il ne devrait. Dans un tel cas, l'administrateur peut souhaiter désactiver l'usage de tels noms de groupe en réglant /nacm/enable-external-groups à "faux".

Certains des nœuds de données lisibles dans ce module YANG peuvent être considérés comme sensibles ou vulnérables dans certains environnements de réseau. Il est donc important de contrôler l'accès en lecture (par exemple, via get, get-config, ou notification) à ces nœuds de données. Voici les sous arborescences et les nœuds de données avec leur sensibilité/vulnérabilité :
- o /nacm/enable-nacm
- o /nacm/read-default
- o /nacm/write-default
- o /nacm/exec-default
- o /nacm/enable-external-groups
- o /nacm/groups
- o /nacm/rule-list

Un administrateur doit interdire l'accès en lecture aux objets mentionnées ci-dessus dans ce modèle de données, car ils révèlent la configuration de contrôle d'accès qui pourrait être considérée comme sensible.

5.2 Questions générales de configuration

Il existe un risque que l'invocation d'opérations de protocole non standard ait des effets collatéraux non documentés. Un administrateur doit construire des règles de contrôle d'accès de façon telle que le magasin de données de configuration soit protégé contre de tels effets collatéraux.

Il est possible qu'une session avec des accès en écriture (par exemple, il est permis d'invoquer <edit-config>), mais sans aucun accès à une sous arborescence de magasin de données particulière contenant des données sensibles, détermine la présence ou la non présence de telles données. Cela peut être fait en produisant de façon répétée une sorte de demande d'édition (créer, mettre à jour, ou supprimer) et éventuellement de recevoir des erreurs "accès refusé" en réponse. Ces attaques de "pêche à la ligne" peuvent identifier la présence ou la non présence de données sensibles spécifiques même sans que le champ "error-path" soit présent dans la réponse <rpc-error>.

Il est possible que l'ensemble des capacités NETCONF change sur le serveur au fil du temps. Si il en est ainsi, il y a alors un risque que de nouvelles opérations de protocole, notifications, et/ou contenus de magasin de données aient été ajoutés sur l'appareil. Un administrateur doit être sûr que les règles de contrôle d'accès sont correctes pour le nouveau contenu. Les mécanismes pour détecter les changements de capacités NETCONF sur un appareil spécifique sortent du domaine

d'application de ce document.

Il est possible que la définition du modèle de données elle-même (par exemple, un when-stmt YANG) aide une session non autorisée à déterminer la présence ou même la valeur de nœuds de données sensibles en examinant la présence et les valeurs de différents nœuds de données.

Il est possible que la définition du modèle de données elle-même (par exemple, un when-stmt ou choice-stmt YANG) permette à une session de créer ou supprimer implicitement des nœuds sur lesquels la session n'a pas d'accès en écriture comme effet collatéral implicite du traitement d'une opération <edit-config> permise.

Il y a un risque que des opérations de protocole non standard, ou même l'opération de protocole standard <get>, puissent retourner des données qui "reproduisent" ou "copient" des données sensibles d'un objet de données différent. Il peut simplement y avoir plusieurs définitions de modèle de données qui exposent ou même configurent la même instrumentation de système sous-jacent.

Un modèle de données peut contenir des clés externes (par exemple, YANG leafref) qui exposent des valeurs provenant d'une structure de données différente. U administrateur doit connaître les modèles de données sensibles qui contiennent des nœuds leafref. Cela englobe de trouver tous les objets leafref qui "pointent" sur les données sensibles (c'est-à-dire, les valeurs "path-stmt") qui implicitement ou explicitement incluent des nœuds de données sensibles.

Il sort du domaine d'application du présent document de définir les procédures d'application de contrôle d'accès pour l'instrumentation de l'appareil sous-jacent qui peut exister pour prendre en charge le fonctionnement du serveur NETCONF. Un administrateur peut identifier chaque opération de protocole que le serveur fournit et décider si il a besoin de lui appliquer un contrôle d'accès.

Le présent document incorpore l'utilisation facultative d'un mécanisme de session de récupération, qui peut être utilisé pour outrepasser l'application du contrôle d'accès en cas d'urgence comme des erreurs de configuration du NACM qui désactivent tous les accès au serveur. La configuration et l'identification d'un tel mécanisme de session de récupération sont spécifiques de la mise en œuvre et sortent du domaine du présent document. Un administrateur doit connaître tous les mécanismes de session de récupération disponibles sur l'appareil et s'assurer qu'ils sont utilisés de façon appropriée.

Il est possible qu'une session interrompe la gestion de la configuration, même sans accès en écriture à la configuration, en verrouillant le magasin de données. Cela peut être fait pour s'assurer que tout ou partie de la configuration reste stable pendant qu'elle est restituée, ou cela peut être fait dans une attaque de "déni de service". Le serveur n'a aucun moyen de voir la différence. Un administrateur peut souhaiter interdire l'accès "exec" aux opérations de protocole suivantes :

- <lock> *(verrouillage)*
- <unlock>*(déverrouillage)*
- <partial-lock>*(verrouillage partiel)*
- <partial-unlock>*(déverrouillage partiel)*

5.3 Considérations sur la conception du modèle de données

Les concepteurs doivent identifier clairement toutes les données sensibles, les notifications, ou les opérations de protocole définies au sein d'un module YANG. Pour une telle définition, une déclaration "nacm:default-deny-write" ou "nacm:default-deny-all" devrait être présente, en plus d'une description claire des risques pour la sécurité.

Les opérations de protocole doivent être bien documentées par le concepteur du modèle de données afin que soit clair pour les administrateurs quels nœuds de données (s'il en est) sont affectés par l'opération de protocole et quelles informations (s'il en est) sont retournées dans le message <rpc-reply>.

Les modèles de données devraient être conçus de façon telle que différents niveaux d'accès pour les paramètres d'entrée aux opérations de protocole ne soient pas nécessaires. L'utilisation d'opérations de protocole génériques devrait être évité, et si différents niveaux d'accès sont nécessaires, des opérations de protocole séparées devraient plutôt être définies.

6. Références

6.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174). DOI 10.17487/RFC2119

[RFC5246] T. Dierks, E. Rescorla, "Version 1.2 du protocole de sécurité de la couche Transport (TLS)", août 2008. *(P.S. ; rendue obsolète par la* RFC8446), DOI 10.17487/RFC5246.

[RFC5277] S. Chisholm, H. Trevino, "Notification d'événements NETCONF", juillet 2008. *(P.S.)*, DOI 10.17487/RFC5277.

[RFC6020] M. Bjorklund, "YANG - Langage de modélisation des données pour le protocole de configuration de réseau (NETCONF)", octobre 2010. *P.S.)*, DOI 10.17487/RFC6020.

[RFC6241] R. Enns, M. Bjorklund, J. Schoenwaelder, A. Bierman, "Protocole de configuration de réseau (NETCONF)", juin 2011. *(Remplace la* RFC4741) *(P.S. ; MàJ par* RFC7803), DOI 10.17487/RFC6241.

[RFC6242] M. Wasserman, "Utilisation du protocole NETCONF sur Secure Shell (SSH)", juin 2011. *(Remplace la* RFC4742) *(P.S.)*, DOI 10.17487/RFC6242.

[RFC6991] J. Schoenwaelder, "Types de données courants dans YANG", juillet 2013. DOI 10.17487/RFC6991. *(P.S.; Remplace* RFC6021).

[RFC7230] R. Fielding, et J. Reschke, "Protocole de transfert Hypertexte (HTTP/1.1) : syntaxe et acheminement de message", juin 2014. *(P.S.)*, DOI 10.17487/RFC7230.

[RFC7950] M. Bjorklund, "Langage de modélisation de données YANG 1.1", août 2016. *(P.S.)*, DOI 10.17487/RFC7950.

[RFC8040] A. Bierman, et autres, "Protocole RESTCONF", janvier 2017. *(P.S.)*, DOI 10.17487/RFC8040.

[RFC8174] B. Leiba, "Ambiguïté des mots clés en majuscules ou minucules dans la RFC2119", mai 2017. BCP14. *(MàJ* RFC2119), DOI 10.17487/RFC8174.

[RFC8342] M. Bjorklund, J. Schoenwaelder, P. Shafer, K. Watsen, R. Wilton, "Architecture de magasin de données de gestion de réseau (NMDA)", mars 2018, DOI 10.17487/RFC8342.

[XML] Bray, T., Paoli, J., Sperberg-McQueen, M., Maler, E., and F. Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", World Wide Web Consortium Recommendation REC-xml-20081126, novembre 2008, <https://www.w3.org/TR/2008/REC-xml-20081126 >.

6.2 Références pour information

[RFC2865] C. Rigney et autres, "Service d'authentification à distance de l'utilisateur appelant (RADIUS)", juin 2000. (*MàJ par* RFC2868, RFC3575, RFC5080, RFC8044) *(D.S.)*, DOI 10.17487/RFC2865.

[RFC5607] D. Nelson, G. Weber, "Autorisation du service d'authentification à distance de l'usager appelant (RADIUS) pour la gestion de serveur d'accès réseau (NAS)", juillet 2009. *(P. S.)*, DOI 10.17487/RFC5607.

[YANG-SEC] IETF, "YANG Security Guidelines", <https://trac.ietf.org/trac/ops/wiki/yang-security-guidelines>.

Appendice A. Exemples d'utilisation

Les courts extraits XML [XML] suivants ne sont fournis qu'à titre d'exemples, pour montrer comment le NACM peut être configuré pour effectuer des tâches de contrôle d'accès.

A.1 Exemple de <groups>

Il doit y avoir au moins une entrée <group> afin qu'une des règles de contrôle d'accès soit utile.

Le XML suivant montre des groupes arbitraires et n'est pas destiné à représenter un cas d'utilisation particulier.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <groups>
```

```
    <group>
      <name>admin</name>
      <user-name>admin</user-name>
      <user-name>andy</user-name>
    </group>

    <group>
      <name>limited</name>
      <user-name>wilma</user-name>
      <user-name>bam-bam</user-name>
    </group>

    <group>
      <name>guest</name>
      <user-name>guest</user-name>
      <user-name>guest@example.com</user-name>
    </group>
  </groups>
</nacm>
```

Cet exemple montre trois groupes :

admin : le groupe "admin" contient deux utilisateurs nommés "admin" et "andy".

limited : le groupe "limited" contient deux utilisateurs nommés "wilma" et "bam-bam".

guest : le groupe "guest" contient deux utilisateurs nommés "guest" et "guest@example.com".

A.2 Exemple de règle de module

Des règles de module sont utilisées pour contrôler l'accès à tout le contenu défini dans un module spécifique. Une règle de module a la feuille "module-name" établie mais aucun nœud du choix "rule-type" n'est établi.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>guest-acl</name>
    <group>guest</group>
    <rule>
      <name>deny-ncm</name>
      <module-name>ietf-netconf-monitoring</module-name>
      <access-operations>*</access-operations>
      <action>deny</action>
      <comment>
        Ne permettre aucun accès des invités aux informations de surveillance NETCONF.
      </comment>
    </rule>
  </rule-list>

  <rule-list>
    <name>limited-acl</name>
    <group>limited</group>
    <rule>
      <name>permit-ncm</name>
      <module-name>ietf-netconf-monitoring</module-name>
      <access-operations>read</access-operations>
      <action>permit</action>
      <comment>
        Permettre l'accès en lecture aux informations de surveillance NETCONF.
      </comment>
    </rule>
    <rule>
      <name>permit-exec</name>
```

```
      <module-name>*</module-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
      <comment>
        Permettre l'invocation des opérations de serveur prises en charge.
      </comment>
    </rule>
  </rule-list>

  <rule-list>
    <name>admin-acl</name>
    <group>admin</group>
    <rule>
      <name>permit-all</name>
      <module-name>*</module-name>
      <access-operations>*</access-operations>
      <action>permit</action>
      <comment>
        Permettre au groupe 'admin' d'accéder à toutes les opérations et données.
      </comment>
    </rule>
  </rule-list>
</nacm>
```

Cet exemple montre quatre règles de module :

deny-ncm : cette règle empêche le groupe "guest" de lire les informations de surveillance dans le module YANG "ietf-netconf-monitoring".

permit-ncm : cette règle permet au groupe "limited" de lire le module YANG "ietf-netconf-monitoring".

permit-exec : cette règle permet au groupe "limited" d'invoquer toute opération de protocole prise en charge par le serveur.

permit-all : cette règle permet au groupe "admin" de réaliser l'accès à tous les contenus du serveur. Aucune règle suivante ne va correspondre pour le groupe "admin" à cause de cette règle de module.

A.3 Exemple de règle d'opération de protocole

Les règles d'opération de protocole sont utilisées pour contrôler l'accès à une opération de protocole spécifique.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>guest-limited-acl</name>
    <group>limited</group>
    <group>guest</group>
    <rule>
      <name>deny-kill-session</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>kill-session</rpc-name>
      <access-operations>exec</access-operations>
      <action>deny</action>
      <comment>
        Ne pas permettre au groupe 'limited' ou au groupe 'guest' de tuer une autre session.
      </comment>
    </rule>
    <rule>
      <name>deny-delete-config</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>delete-config</rpc-name>
      <access-operations>exec</access-operations>
      <action>deny</action>
      <comment>
        Ne pas permettre au groupe 'limited' ou au groupe 'guest' de supprimer une configuration.
      </comment>
    </rule>
  </rule-list>
```

```
  <rule-list>
    <name>limited-acl</name>
    <group>limited</group>
    <rule>
      <name>permit-edit-config</name>
      <module-name>ietf-netconf</module-name>
      <rpc-name>edit-config</rpc-name>
      <access-operations>exec</access-operations>
      <action>permit</action>
      <comment>
        Permettre au groupe 'limited' d'éditer la configuration.
      </comment>
    </rule>
  </rule-list>
</nacm>
```

Cet exemple montre trois règles d'opération de protocole :

deny-kill-session : cette règle empêche le groupe "limited" ou le groupe "guest" d'invoquer l'opération de protocole NETCONF <kill-session>.

deny-delete-config : cette règle empêche le groupe "limited" ou le groupe "guest" d'invoquer l'opération de protocole NETCONF <delete-config>.

permit-edit-config : cette règle permet au groupe "limited" d'invoquer l'opération de protocole NETCONF <edit-config>. Cette règle n'aura pas d'effet réel sauf si la feuille "exec-default" est réglée à "deny".

A.4 Exemple de règle de nœud de données

Les règles de nœud de données sont utilisées pour contrôler l'accès à des nœuds de données spécifiques (config et non config) au sein du contenu NETCONF fourni par le serveur.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>guest-acl</name>
    <group>guest</group>
    <rule>
      <name>deny-nacm</name>
      <path xmlns:n="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
        /n:nacm
      </path>
      <access-operations>*</access-operations>
      <action>deny</action>
      <comment>
        Refuser au groupe 'guest' tout accès aux données /nacm.
      </comment>
    </rule>
  </rule-list>

  <rule-list>
    <name>limited-acl</name>
    <group>limited</group>
    <rule>
      <name>permit-acme-config</name>
      <path xmlns:acme="http://example.com/ns/netconf">
        /acme:acme-netconf/acme:config-parameters
      </path>
      <access-operations>
        read create update delete
      </access-operations>
      <action>permit</action>
      <comment>
        Permettre au groupe 'limited' un accès complet aux paramètres de configuration acme NETCONF. Montrant la forme
        longue de 'access-operations' au lieu de l'abrégée.
```

```
      </comment>
    </rule>
  </rule-list>

  <rule-list>
    <name>guest-limited-acl</name>
    <group>guest</group>
    <group>limited</group>
    <rule>
      <name>permit-dummy-interface</name>
      <path xmlns:acme="http://example.com/ns/itf">
        /acme:interfaces/acme:interface[acme:name='dummy']
      </path>
      <access-operations>read update</access-operations>
      <action>permit</action>
      <comment>
        Permettre aux groupes 'limited' et 'guest' l'accès en lecture et mise à jour à l'interface factice.
      </comment>
    </rule>
  </rule-list>

  <rule-list>
    <name>admin-acl</name>
    <group>admin</group>
    <rule>
      <name>permit-interface</name>
      <path xmlns:acme="http://example.com/ns/itf">
        /acme:interfaces/acme:interface
      </path>
      <access-operations>*</access-operations>
      <action>permit</action>
      <comment>
        permettre au groupe 'admin' le plein accès à toutes les interfaces acme.
      </comment>
    </rule>
  </rule-list>
</nacm>
```

Cet exemple montre quatre règles de nœud de données :
deny-nacm : cette règle refuse au groupe "guest" tout accès à la sous arborescence /nacm.
permit-acme-config : cette règle donne au groupe "limited" l'accès en lecture-écriture aux <config-parameters> acme.
permit-dummy-interface : cette règle donne aux groupes "limited" et "guest" l'accès en lecture-mise à jour à l'entrée <interface> acme nommée "dummy". Cette entrée ne peut pas être créée ou supprimée par ces groupes ; elle peut seulement être altérée.
permit-interface : cette règle donne au groupe "admin" l'accès en lecture-écriture à toutes les entrées <interface> acme.

A.5 Exemple de règle de notification

Les règles de notification sont utilisées pour contrôler l'accès à un type spécifique d'événement de notification.

```
<nacm xmlns="urn:ietf:params:xml:ns:yang:ietf-netconf-acm">
  <rule-list>
    <name>sys-acl</name>
    <group>limited</group>
    <group>guest</group>      <rule>
      <name>deny-config-change</name>
      <module-name>acme-system</module-name>
      <notification-name>sys-config-change</notification-name>
      <access-operations>read</access-operations>
      <action>deny</action>
      <comment>
        Ne pas permettre au groupe 'guest' ou au groupe 'limited' de recevoire des événements de changement de
```

```
        configuration.
      </comment>
    </rule>
  </rule-list>
</nacm>
```

Cet exemple montre une règle de notification : deny-config-change. Cette règle empêche les groupes "limited" ou "guest" de recevoir le type d'événemente acme <sys-config-change>.

Adresse des auteurs

Andy Bierman
YumaWorks
685 Cochran St.
Suite #160
Simi Valley, CA 93065 USA
mél : andy@yumaworks.com

Martin Bjorklund
Tail-f Systems
mél : mbj@tail-f.com