Groupe de travail Réseau
Request for Comments : 5025
Catégorie : Sur la voie de la normalisation

J. Rosenberg, Cisco
décembre 2007
Traduction Claude Brière de L'Isle

Règles d'autorisation de présence

Résumé
L'autorisation est une fonction clé dans les systèmes de présence. Les politiques d'autorisation, aussi appelées règles d'autorisation, spécifient quelles informations de présence peuvent être données à quels observateurs, et quand. La présente spécification définit un format de document en langage de balisage extensible (XML, *Extensible Markup Language*) pour exprimer les règles d'autorisation de présence. Un tel document peut être manipulé par les clients en utilisant le protocole d'accès de configuration XML (XCAP, *XML Configuration Access Protocol*) bien que d'autres techniques soient permises.

Table des matières

1. Introduction

Les spécifications du protocole d'initialisation de session (SIP, *Session Initiation Protocol*) pour la messagerie instantanée

et la présence (SIMPLE, *SIP for Instant Messaging et Presence*) permettent à un utilisateur, appelé un observateur, de s'abonner à un autre utilisateur, appelé une présentité [RFC2778], afin d'apprendre ses informations de présence [RFC3856]. Cet abonnement est traité par un agent de présence. Cependant, les informations de présence sont sensibles, et un agent de présence a besoin d'une autorisation de la présentité avant de traiter les informations de présence. À ce titre, un format de document d'autorisation de présence est nécessaire. La présente spécification définit un format pour ce document, appelé un document d'autorisation de présence.

La [RFC4745] spécifie un cadre pour représenter les politiques d'autorisation, et est applicable aux systèmes comme la géolocalisation et la présence. Ce cadre est utilisé comme base des documents d'autorisation de présence. Dans ce cadre, une politique d'autorisation est un ensemble de règles. Chaque règle contient des conditions, des actions, et des transformations. Les conditions spécifient sous quelles conditions la règle va être appliquée au traitement du serveur de présence. L'élément d'action dit au serveur quelles actions entreprendre. L'élément de transformation indique comment les données de présence vont être manipulées avant d'être présentées à cet observateur, et à ce titre, définit une opération de filtrage de confidentialité. La [RFC4745] identifie un petit nombre de conditions spécifiques communes aux services de présence et de localisation, et laisse à d'autres spécifications, comme la présente, de compléter les détails spécifiques d'usage.

Un document d'autorisation de présence peut être manipulé par les clients en utilisant plusieurs moyens. Un de ces mécanismes est le protocole d'accès à la configuration XML (XCAP, *XML Configuration Access Protocol*) [RFC4825]. La présente spécification définit les détails nécessaires pour utiliser XCAP pour gérer les documents d'autorisation de présence.

2. 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 la [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

3. Structure des documents d'autorisation de présence

Un document d'autorisation de présence est un document XML, formaté conformément au schéma défini dans la [RFC4745]. Les documents d'autorisation de présence héritent du type MIME des documents de politique commune, application/auth-policy+xml. Comme décrit dans la [RFC4745], le présent document est composé de règles qui contiennent trois parties - conditions, actions, et transformations. Chaque action ou transformation, qui est aussi appelée une permission, a la propriéte d'être un octroi positif d'informations à l'observateur. Par suite, il y a un mécanisme bien défini pour combiner les actions et transformations obtenues de plusieurs sources. Ce mécanisme préserve la confidentialité, car l'absence de toute action ou transformation peut seulement résulter en moins d'informations présentées à un observateur.

Cette Section définit les nouvelles conditions, actions, et transformations définies par la présente spécification.

3.1 Conditions

3.1.1 Identité

Bien que l'élément <identity> soit défini dans la [RFC4745], la présente spécification indique les usages spécifiques du document cadre qui doivent définir les détails qui sont spécifiques du protocole et de l'usage. En particulier, il est nécessaire pour un usage du cadre de politique commun de :

- définir les moyens acceptables d'authentification,
- définir la procédure pour représenter l'identité du demandeur/observateur comme un URI ou un identifiant de ressource internationalisé (IRI, *Internationalized Resource Identifier*) [RFC3987].

Ce paragraphe définit les détails pour les systèmes fondés sur la [RFC3856]. Il le fait en termes généraux, de sorte que les recommandations définies ici s'appliquent aux mécanismes existants et futurs d'authentification dans SIP.

3.1.1.1 Formes acceptables d'authentification

Quand elle est utilisée avec SIP, une demande est considérée comme authentifiée si une des conditions suivantes est vraie :

L'observateur prouve son identité au serveur par une forme d'authentification cryptographique, incluant l'authentification fondée sur un secret partagé ou un certificat, et l'authentification donne une identité pour l'observateur.

La demande vient d'un envoyeur qui affirme l'identité de l'observateur, et :
1. l'assertion inclut l'affirmation que la partie qui fait l'assertion a utilisé une forme d'authentification cryptographique (comme définie ci-dessus) pour déterminer l'identité de l'observateur, et
2 le serveur fait confiance à cette assertion, et
3. l'assertion fournit une identité sous la forme d'un URI.

Sur la base de cette définition, des exemples de techniques valides d'authentification incluent SIP [RFC3261], l'authentification par résumé [RFC2617], les assertions d'identité vérifiées cryptographiquement [RFC4474], et les assertions d'identité faites dans des environnements de réseau clos [RFC3325].

Cependant, l'authentification anonyme décrite au paragraphe 22.1 de la [RFC3261] n'est pas considérée comme un mécanisme valide pour l'authentification parce que elle ne produit pas une identité pour l'observateur. Cependant, un champ d'en-tête From anonyme, quand il est utilisé en conjonction avec la [RFC4474], est considéré être un mécanisme acceptable pour l'authentification, car il implique quand même que le nœud qui fait l'assertion a effectué l'authentification qui a produit l'identité de l'observateur.

3.1.1.2 Calcul d'un URI pour l'observateur

Calculer l'URI pour l'observateur dépend de si l'identité est certifiée par l'authentification ou par une identité affirmée.

Si une assertion d'identité est utilisée, l'identité affirmée elle-même (qui est sous la forme d'un URI pour les formes acceptables d'assertion d'identité) est utilisée comme URI. Si le mécanisme d'assertion d'identité affirme plusieurs URI pour l'observateur, alors chacun d'eux est utilisé pour les comparaisons mentionnées dans la [RFC4745], et si l'un d'eux correspond à un élément <one> ou <except>, l'observateur est considéré correspondre.

Si une identité est déterminée directement par une authentification cryptographique, cette authentification doit produire un URI, ou doit produire une forme d'identifiant qui peut être relié, par provisionnement, à un URI lié à cet identifiant.

Par exemple, dans le cas de l'authentification SIP par résumé, le processus d'authentification produit un nom d'utilisateur dont la portée est dans un domaine. Ce nom d'utilisateur et le domaine sont liés à une adresse d'enregistrement (AOR, *Address of Record*) par provisionnement, et l'AOR résultante est utilisée comme URI de l'observateur. Considérons "l'enregistrement d'utilisateur" suivant dans une base de données :

SIP AOR: sip:alice@exemple.com
digest username: ali
digest password: f779ajvvh8a6s6
digest realm: exemple.com

Si le serveur de présence reçoit une demande SUBSCRIBE, il la confronte au domaine réglé à "exemple.com", et le SUBSCRIBE suivant contient un champ d'en-tête Authorization avec un nom d'utilisateur de "ali" et une réponse de résumé générée avec le mot de passe "f779ajvvh8a6s6", l'identité utilisée dans les opérations de confrontation est "sip:alice@exemple.com".

Dans les systèmes SIP, il est possible à un utilisateur d'avoir des alias - c'est-à-dire, il y a plusieurs AOR SIP "allouées" à un seul utilisateur. Dans les termes de la présente spécification, il n'y a pas de relation entre ces alias. Chacun va sembler un utilisateur différent. Ce sera la conséquence pour les systèmes où l'observateur est dans un domaine différent de celui de la présentité. Cependant, même si l'observateur et la présentité sont dans le même domaine, et si le serveur de présence sait que ce sont des alias de l'observateur, ces alias ne sont pas transposés l'un en l'autre ou utilisés d'une autre façon.

SIP permet aussi des demandes anonymes. Si une demande est anonyme parce que l'observateur a utilisé un mécanisme d'authentification qui ne donne pas une identité au serveur de présence (comme le nom d'utilisateur de résumé SIP "anonymous") la demande est considérée non authentifiée (comme expliqué plus haut) et va seulement correpondre à un élément <identity> vide. Si une demande est anonyme parcc qu'elle contient un champ d'en-tête Privacy [RFC3323], mais contient quand même une identité affirmée satisfaisant aux critères définis ci-dessus, cette identité est utilisée, et le fait que

la demande était anonyme n'a pas d'impact sur le traitement d'identité.

Il est important de noter que SIP utilise fréquemment à la fois des URI SIP et des URI tel [RFC3966] comme identifiants, et pour rendre les choses encore plus confuses, un URI SIP peut contenir un numéro de téléphone dans sa partie utilisateur, dans le même format qu'utilisé dans un URI tel. Une identité de demandeur/observateur qui est un URI SIP avec un numéro de téléphone NE va PAS correspondre aux conditions <one> et <except> dont l'identifiant est un URI tel avec le même numéro. L'inverse est aussi vrai. Si l'identité de demandeur/observateur est un URI tel, cela ne va pas correspondre à un URI SIP dans les conditions <one> ou <except> dont la partie utilisateur est un numéro de téléphone. Les URI de schémas différents ne sont jamais équivalents.

3.1.2 Sphère

L'élément <sphere> est défini dans la [RFC4745]. Cependant, chaque application qui utilise la spécification de politique commune a besoin de déterminer comment le serveur de présence calcule la valeur de la <sphere> à utiliser dans l'évaluation de la condition.

Pour calculer la valeur de <sphere>, l'agent de présence examine tous les documents de présence publiés pour la présentité : si au moins un d'eux inclut l'élément <sphere> [RFC4480] au titre du composant de données de personne [RFC4479], et si tous ceux qui contiennent l'élément ont la même valeur pour lui, qui est la valeur utilisée pour la <sphere> dans le traitement de politique de présence. Si, cependant, l'élément <sphere> n'est présent dans aucun des documents publiés, ou si il est présent mais a des valeurs incohérentes, sa valeur est considérée comme indéfinie en termes de traitement de politique de présence.

Il faut faire attention quand on utilise <sphere> comme condition pour déterminer le traitement de l'abonnement. Comme la valeur de <sphere> change de façon dynamique, un changement d'état peut causer une terminaison soudaine de l'abonnement. L'observateur n'a pas de moyen de savoir, sauf en interrogeant, quand son abonnement va être réinstallé lorsque la valeur de <sphere> change. Pour cette raison, <sphere> est principalement utile pour la confrontation aux règles qui définissent des transformations.

3.2 Actions

3.2.1 Traitement de l'abonnement

L'élément <sub-handling> *(sous traitement)* spécifie la décision d'autorisation d'abonnement que le serveur devrait faire. Il spécifie aussi si le document de présence pour l'observateur devrait ou non être construit en utilisant un "polite blocking" *(blocage poli)*. L'usage du blocage poli et la décision d'autorisation d'abonnement sont spécifiés conjointement car le traitement de confidentialité approprié exige une corrélation entre eux. Comme discuté dans la [RFC4745], comme l'algorithme de combinaison fonctionne indépendamment pour chaque permission, si une corrélation existe entre les permissions, elles doivent être fusionnées en une seule variable. C'est ce qui est fait ici. L'élément <sub-handling> est de type entier énuméré. Les valeurs définies sont :

block *(bloque)* : cette action dit au serveur de rejeter l'abonnement, le plaçant dans l'état "terminé". Il a la valeur de zéro, et il représente la valeur par défaut. Aucune valeur de l'élément <sub-handling> ne peut jamais être inférieure à celle là. Strictement parlant, il n'est pas nécessaire qu'une règle inclue une action de blocage explicite, car par défaut en l'absence de toute action elle va être bloquée. Cependant, elle est incluse pour être complet.

confirm *(confirme)* : cette action dit au serveur de placer l'abonnement dans l'état "en cours" *(pending)*, et d'attendre une entrée de la présentité pour déterminer comment procéder. Elle a une valeur de dix.

polite-block *(blocage poli)* : cette action dit au serveur de placer l'abonnement dans l'état "actif", et de produire un document de présence qui indique que la présentité est indisponible. Un document raisonnable excluerait les éléments d'information d'appareil et de personne, et inclurait seulement un service dont l'état de base est réglé à fermé [RFC3863]. Cette action a une valeur de vingt.

allow *(permet)* : cette action dit au serveur de placer l'abonnement dans l'état "actif". Cette action a une valeur de trente.

Note : placer une valeur de blocage pour cet élément ne garantit pas qu'un abonnement est refusé ! Si une règle de correspondance a une autre valeur pour cet élément, l'abonnement va recevoir un traitement fondé sur le maximum de ces autres valeurs. Ceci se fonde sur les règles de combinaison définies dans la [RFC4745].

Les futures spécifications qui souhaiteraient définir de nouveaux types d'actions DOIVENT définir une action entièrement nouvelle (séparée de <sub-handling>) et définir leur propre ensemble de valeurs pour cette action. Un document pourrait contenir à la fois <sub-handling> et une action de traitement d'abonnement définie par une future spécification ; dans ce cas, comme chaque action est toujours une attribution positive d'informations, l'action résultante est la moins restrictive des deux éléments.

Le comportement exact d'un serveur de présence lors d'un changement de valeur de sous traitement peut être décrit en utilisant l'automate à états de traitement d'abonnement de la Figure 1 de la [RFC3857]. Si la permission <sub-handling> change sa valeur à "block", cela cause la génération d'un événement "rejeté" dans l'automate à états d'abonnement pour tous les abonnements affectés. Cela va causer le passage de l'automate à états à l'état "terminé", résultant en la transmission d'un NOTIFY à l'observateur avec un champ d'en-tête État d'abonnement de valeur "terminé" et une raison de "rejeté" [RFC3265], qui termine leur abonnement. Si un nouvel abonnement arrive plus tard, et si la valeur de <sub-handling> qui s'applique à cet abonnement est "block", le traitement d'abonnement suit la branche "subscribe, policy=reject" provenant de l'état "init", et une réponse 403 au SUBSCRIBE est générée.

Si la permission <sub-handling> change sa valeur à "confirm", le traitement dépend des états des abonnements affectés. Malheureusement, l'automate à états dans la RFC 3857 ne définit pas d'événement correspondant à une décision d'autorisation de "en cours". Si l'abonnement est dans l'état "actif", il revient à l'état "en cours". Cela cause l'envoi d'un NOTIFY, mettant à jour l'état d'abonnement [RFC3265] à "en cours". Aucune raison n'est incluse dans le champ d'en-tête État d'abonnement (aucun n'est défini pour traiter ce cas). Aucun autre document n'est envoyé à cet observateur. Il n'y a pas de changement d'état si l'abonnement est dans les états "en cours", "en attente", ou "terminé". Si un nouvel abonnement arrive plus tard, et si la valeur de <sub-handling> qui s'applique à cet abonnement est "confirm", le traitement d'abonnement suit la branche "subscribe, no policy" provenant de l'état "init", et une réponse 202 au SUBSCRIBE est générée, suivie par un NOTIFY avec l'état d'abonnement de "en cours". Aucun document de présence n'est inclus dans ce NOTIFY.

Si la permission <sub-handling> change de valeur de "blocked" ou "confirm" à "polite-block" ou "allow", cela cause un événement "approved" généré dans l'automate à états pour tous les abonnements affectés. Si l'abonnement était dans l'état "en cours", l'automate à états va passer à l'état "actif", résultant en la transmission d'un NOTIFY avec un champ d'en-tête État d'abonnement de "actif", et l'inclusion d'un document de présence dans ce NOTIFY.

Si l'abonnement était dans l'état "en attente", il va passe à l'état "terminé". Si un nouvel abonnement arrive plus tard, et si la valeur de <sub-handling> qui s'applique à cet abonnement est "polite-block" ou "allow", le traitement d'abonnement suit la branche "subscribe, policy=accept" provenant de l'état "init", et une réponse 200 OK au SUBSCRIBE est générée, suivie par un NOTIFY avec l'état d'abonnement de "actif" avec un document de présence dans le corps du NOTIFY.

3.3 Transformations

Les transformations définies ici sont utilisées pour piloter le comportement de l'opération de filtrage de confidentialité. Chaque transformation qui définit la visibilité d'un observateur est accordée à un composant particulier du document de présence. Un groupe de transformations accorde la visibilité aux éléments de données de personne, appareil, et service sur la base des informations d'identification pour ces éléments. Un autre groupe de transformations fournit l'accès aux éléments de données particuliers dans le document de présence.

3.3.1 Fourniture de l'accès aux éléments de composant de données

Les transformations de ce paragraphe fournissent l'accès aux éléments de composant de données de personne, appareil, et service. Une fois que l'accès a été accordé à un tel élément, l'accès aux attributs de présence spécifiques pour cet élément est contrôlé par les permissions définies au paragraphe 3.3.2.

3.3.1.1 Informations d'appareil

La permission <provide-devices> permet à un observateur de voir les informations <device> présentes dans le document de présence. C'est un ensemble variable. Chaque membre de l'ensemble fournit un moyen d'identifier un appareil ou groupe d'appareils. La présente spécification définit trois types d'éléments dans l'ensemble - <class>, qui identifie une occurrence d'appareil par classe, <deviceID>, qui identifie une occurrence d'appareil par l'identifiant d'appareil, et <occurrence-id>, qui identifie une occurrence d'appareil par identifiant d'occurrence. L'identifiant d'appareil et l'identifiant d'occurrence sont définis dans la [RFC4479]. Chaque membre de l'ensemble est identifié par son type (classe, identifiant d'appareil, ouidentifiant d'occurrence) et sa valeur (valeur de la classe, valeur de l'identifiant d'appareil, ou valeur de l'identifiant

d'occurrence).

Par exemple, considérons l'élément <provide-devices> suivant :

```
<provide-devices>
  <deviceID>urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6</deviceID>
  <class>biz</class>
</provide-devices>
```

Cet ensemble a deux membres. Ceci est combiné avec un élément <provide-devices> provenant d'une règle différente :

```
<provide-devices>
  <class>home</class>
  <class>biz</class>
</provide-devices>
```

Le résultat de la combinaison d'ensembles (en utilisant l'opération union) est un ensemble avec trois éléments :

```
<provide-devices>
  <class>home</class>
  <class>biz</class>
  <deviceID>urn:uuid:f81d4fae-7dec-11d0-a765-00a0c91e6bf6</deviceID>
</provide-devices>
```

L'élément <provide-devices> peut aussi prendre la valeur particulière <all-devices>, qui est une notation abrégée pour toutes les occurrences d'appareils présentes dans le document de présence.

La permission est accordée de voir une occurrence d'appareil particulière si un des identifiants d'appareil dans l'ensemble identifie cette occurrence d'appareil. Si une permission <class> est accordée à l'observateur, et si la <class> de l'occurrence d'appareil correspond à la valeur de la permission <class> sur la base d'une égalité sensible à la casse, l'occurrence d'appareil est incluse dans le document de présence. Si une permission <deviceID> est accordée à l'observateur, et si le <deviceID> de l'occurrence d'appareil correspond à la valeur de la permission <deviceID> sur la base d'une équivalence d'URI, l'occurrence d'appareil est incluse dans le document de présence. Si une permission <occurrence-id> est accordée à l'observateur, et si <occurrence-id> de l'occurrence d'appareil correspond à la valeur de la permission <occurrence-id> sur la base d'une égalité sensible à la case, l'occurrence d'appareil est incluse dans le document de présence. De plus, une occurrence d'appareil est incluse dans le document de présence si la permission <all-devices> a été accordée à l'observateur.

3.3.1.2 Informations de personne

La permission <provide-persons> permet à un observateur de voir les informations de <person> présentes dans le document de présence. C'est un ensemble variable. Chaque membre de l'ensemble fournit un moyen d'identifier une occurrence de personne. La présente spécification définit deux types d'éléments dans l'ensemble - <class>, qui identifie une occurrence de personne par classe, et <occurrence-id>, qui identifie une occurrence par son identifiant d'occurrence. Chaque membre de l'ensemble est identifié par son type (classe ou identifiant d'occurrence) et sa valeur (valeur de la classe ou valeur de l'identifiant d'occurrence). L'élément <provide-persons> peut aussi prendre la valeur spéciale de <all-persons>, qui est une notation abrégée pour toutes les occurrences de personnes présentes dans le document de présence. La combinaison d'ensembles est identique à celle de l'élément <provide-devices>.

La permission est accordée de voir une occurrence de personne particulière si un des identifiants de personne dans l'ensemble identifie cette occurrence de personne. Si une permission <class> est accordée à l'observateur, et si la <class> de l'occurrence de personne correspond à la valeur de la permission <class> sur la base d'une égalité sensible à la casse, l'occurrence de personne est incluse dans le document de présence. Si une permission <occurrence-id> est accordée à l'observateur, et si l'<occurrence-id> de l'occurrence de personne correspond à la valeur de la permission <occurrence-id> sur la base d'une égalité sensible à la casse, l'occurrence de personne est incluse dans le document de présence. De plus, une occurrence de personne est incluse dans le document de présence si la permission <all-persons> a été accordée à l'observateur.

3.3.1.3 Informations de service

La permission <provide-services> permet à un observateur de voir les informations de service présentes dans les éléments <tuple> du document de présence. Comme <provide-devices>, c'est un ensemble variable. Chaque membre de l'ensemble fournit un moyen pour identifier une occurrence de service. La présente spécification définit quatre types d'éléments dans l'ensemble - <class>, qui identifie une occurrence de service par classe, <occurrence-id>, qui identifie un service par son identifiant d'occurrence, <service-uri>, qui identifie un service par son URI de service, et <service-uri-scheme>, qui identifie un service par son schéma d'URI de service. Chaque membre de l'ensemble est identifié par son type (classe, identifiant d'occurrence, uri de service, ou schéma d'uri de service) et sa valeur (valeur de la classe, valeur de l'identifiant d'occurrence, valeur d'uri de service, ou valeur du schéma d'uri de service). L'élément <provide-services> peut aussi prendre la valeur spéciale de <all-services>, qui est une notation abrégée pour toutes les occurrences de service présentes dans le document de présence. La combinaison des ensembles est faite comme pour l'élément <provide-persons>.

La permission est accordée pour voir une occurrence de service particulière si un des identifiants de service dans l'ensemble identifie cette occurrence de service. Si une permission <class> est accordée à l'observateur, et si la <class> de l'occurrence de service correspond à la valeur de la permission <class> sur la base d'une égalité sensible à la casse, l'occurrence de service est incluse dans le document de présence. Si une permission <service-uri> est accordée à l'observateur, et si le <service-uri> de l'occurrence de service correspond à la valeur de la permission <service-uri> sur la base de l'équivalence d'URI, l'occurrence de service est incluse dans le document de présence. Si une permission <occurrence-id> est accordée à l'observateur, et si le <occurrence-id> de l'occurrence de service correspond à la valeur de la permission <occurrence-id> sur la base d'une égalité sensible à la casse, l'occurrence de service est incluse dans le document de présence. Si une permission <service-uri-scheme> est accordée à l'observateur, et si le schéma de l'URI de service pour l'occurrence de service correspond à la valeur de <service-uri-scheme> sur la base d'une égalité sensible à la casse, l'occurrence de service est incluse dans le document de présence. De plus, une occurrence de service est incluse dans le document de présence si la permission <all-services> a été accordée à l'observateur.

3.3.2 Fourniture d'accès aux attributs de présence

Les permissions du paragraphe 3.3.1 fournissent un accès grossier aux données de présence en permettant ou en bloquant des services ou appareils spécifiques, et en permettant ou bloquant des informations de personne.

Une fois que les informations de personne, appareil, ou service sont incluses dans le document, les permissions de ce paragraphe définissent quels attributs de présence y sont rapportés. Certaines informations sont toujours rapportées. En particulier, les éléments <contact>, <service-class> [RFC4480], l'état <basic>, et les éléments <horodatage> dans tous les éléments <tuple>, si présents, sont fournis aux observateurs. L'élément <horodatage> dans tous les éléments <person>, si il est présent, est fourni aux observateurs. Les éléments <horodatage> et <deviceID> dans tous les éléments <device>, si il est présent, sont fournis à tous les observateurs.

3.3.2.1 Fourniture d'activités

Cette permission contrôle l'accès à l'élément <activities> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-activities>, et il est de type Booléen. Si sa valeur est VRAI, l'élément <activities> dans l'élément de données de personne est alors rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est suppimé si il est présent.

3.3.2.2 Fourniture de classe

Cette permission contrôle l'accès à l'élément <class> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-class>, et il est de type Booléen. Si sa valeur est VRAI, alors tout élément <class> dans un élément de données de personne, service, ou appareil est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.3 Fourniture d'identifiant d'appareil

Cette permission contrôle l'accès à l'élément <deviceID> dans un élément <tuple>, comme défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-deviceID>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <deviceID> dans l'élément de données de service est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent. Noter que le <deviceID> dans un élément de données d'appareil est toujours inclus, et n'est pas contrôlé par cette permission.

3.3.2.4 Fourniture de mode

Cette permission contrôle l'accès à l'élément <mood> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-mood>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <mood> dans l'élément de données de personne est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.5 Fourniture d'emplacement

Cette permission contrôle l'accès à l'élément <place-is> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-place-is>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <place-is> dans l'élément de données de personne est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.6 Fourniture de type d'emplacement

Cette permission contrôle l'accès à l'élément <place-type> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-place-type>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <place-type> dans l'élément de données de personne est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.7 Fourniture de confidentialité

Cette permission contrôle l'accès à l'élément <privacy> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-privacy>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <privacy> dans l'élément de données de personne ou service est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.8 Fourniture de relation

Cette permission contrôle l'accès à l'élément <relationship> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-relationship>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <relationship> dans l'élément de données de service est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.9 Fourniture de sphère

Cette permission contrôle l'accès à l'élément <sphere> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-sphere>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <sphere> dans l'élément de données de personne est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.10 Fourniture d'image d'état

Cette permission contrôle l'accès à l'élément <status-icon> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-status-icon>, et il est de type Booléen. Si sa valeur est VRAI, alors tout élément <status-icon> dans l'élément de données de personne ou service est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.11 Fourniture du décalage de temps

Cette permission contrôle l'accès à à l'élément <time-offset> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette permission est <provide-time-offset>, et il est de type Booléen. Si sa valeur est VRAI, alors l'élément <time-offset> dans l'élément de données de personne est rapporté à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

3.3.2.12 Fourniture d'entrée d'utilisateur

Cette permission contrôle l'accès à l'élément <user-input> défini dans la [RFC4480]. Le nom de l'élément qui fournit cette

permission est <provide-user-input>, et il est de type entier énuméré. Sa valeur définit quelles informations sont fournies aux observateurs dans les éléments de données de personne, appareil, ou service:

faux : cette valeur indique que l'élément <user-input> est supprimé du document. Il lui est alloué la valeur numérique de 0.

nu : cette valeur indique que l'élément <user-input> est à conserver. Cependant, tous les attributs "idle-threshold" et "since" sont à supprimer. Il lui est alloué la valeur numérique de 10.

seuils : cette valeur indique que l'élément <user-input> est à conserve. Cependant, seul l'attribut "idle-threshold" est à conserver. Il lui est alloué la valeur numérique de 20.

plein : cette valeur indique que l'élément <user-input> est à conserver, incluant tous les attributs. Il lui est alloué la valeur numérique de 30.

3.3.2.13. Fourniture de note

Cette permission contrôle l'accès à l'élément <note> défini dans la [RFC3863] pour <tuple> et dans la [RFC4479] pour <person> et <device>. Le nom de l'élément qui fournit cette permission est <provide-note>, et il est de type Booléen. Si sa valeur est VRAI, alors tous les éléments <note> dans les éléments de données de personne, service ou appareil sont rapportés à l'observateur. Si c'est FAUX, cet attribut de présence est supprimé si il est présent.

Cette permission n'a pas d'effet sur les valeurs de <note> présentes dans les éléments <activities>, <mood>, <place-is>, <place-type>, <privacy>, <relationship>, ou <service-class>. Les notes dans ces éléments sont essentiellement des valeurs pour leurs éléments respectifs, et sont présentes si l'élément concerné est permis dans le document de présence. Par exemple, si un élément <activities> est présent dans un document de présence, et si il y a une valeur de <note> pour lui, cette note est présente dans le document envoyé à l'observateur si la permission <provide-activities> est donnée, sans considération de si la permisssion <provide-note> est donnée.

3.3.2.14 Fourniture d'attribut inconnu

Il est important qu'il soit permis aux systèmes d'inclure des informations de présence propriétaires ou nouvelles et que les utilisateurs soient capables d'établir des permissions pour ces informations, sans exiger une mise à niveau du serveur de présence et du système d'autorisation. C'est pour cette raison qu'est définie la permission <provide-unknown-attribute>. Cette permission indique que l'attribut de présence inconnu avec le nom et l'espace de noms donnés (fournis comme des attributs obligatoires de l'élément <provide-unknown-attribute>) devrait être inclus dans le document. Son type est Booléen.

La valeur de l'attribut nom DOIT être un nom d'élément non qualifié (ce qui signifie qu'un préfixe d'espace de noms NE DOIT PAS être inclus) et la valeur de l'attribut ns DOIT être un URI d'espace de noms. Les deux sont combinés pour former un nom d'élément qualifié, qui va être confronté à tous les éléments fils inconnus des éléments <tuple>, <device>, ou <person> du format de données d'informations de présence (PIDF, *Presence Information Data Format*) qui ont le même nom qualifié. Dans ce contexte, "inconnu" signifie que le serveur de présence n'a pas connaissance des schémas qui définissent les politiques d'autorisation pour cet élément. Par définition, cela va exclure l'application de la règle <provide-unknown-attribute> à toutes les extensions d'état de présence définies par RPID, car les politiques d'autorisation pour elles sont définies ici.

Une autre conséquence de cette définition est que l'interprétation de l'élément <provide-unknown-attribute> peut changer si le serveur de présence doit être mis à niveau. Par exemple, si on considère un serveur qui, avant la mise à niveau, avait un document d'autorisation qui utilisait <provide-unknown-attribute> avec une valeur de VRAI pour un certain attribut, disons foo. Cet attribut était d'un espace de noms et d'un schéma inconnus du serveur, et donc l'attribut a été fourni aux observateurs. Cependant, après la mise à niveau, le serveur est maintenant au courant d'un nouvel espace de noms et schéma pour une permission qui accorde l'accès à l'attribut foo. Maintenant, la permission <provide-unknown-attribute> pour l'attribut foo va être ignorée, résultant en une suppression de ces éléments des documents de présence envoyés aux observateurs. La confidentialité du système reste sûre, mais le comportement pourrait n'être pas celui attendu. Il est conseillé aux développeurs de systèmes qui permettent aux clients de régler leurs politiques de vérifier les capacités du serveur (en utilisant le mécanisme décrit à la Section 8) avant de télélcharger un nouveau document d'autorisation, pour s'assurer que le comportement va être celui attendu.

3.3.2.15 Fourniture de tous les attributs

Cette permission accorde l'accès à tous les attributs de présence dans tous les éléments de personne, appareil, et tuple présents dans le document (ceux présents dans le document sont déterminés par les permissions <provide-persons>, <provide-devices>, et <provide-services>). C'est effectivement une macro qui s'étend en un ensemble de permissions provide-activities, provide-class, provide-deviceID, provide-mood, provide-place-is, provide-place-type, provide-privacy, provide-relationship, provide-sphere, provide-status-icon, provide-time-offset, provide-user-input, provide-note, et provide-unknown-attribute telle que chaque attribut de présence dans le document ait une permission pour lui. Cela implique que, pour autant qu'une occurrence entière de personne, service, ou appareil est fournie, chaque attribut de présence seul, incluant ceux qui ne sont pas connus du serveur et/ou qui seront définis dans de futures extensions de document de présence, est accordé à l'observateur.

4. Quand appliquer les politiques d'autorisation

La présente spécification ne rend pas obligatoire le point du traitement des données de présence auquel les aspects de filtrage de confidentialité de la politique d'autorisation sont appliqués. Cependant, ils doivent être appliqués de telle sorte que le document de présence final envoyé à l'observateur soit conforme à la politique de confidentialité décrite dans les documents d'autorisation qui s'appliquent à l'utilisateur (il peut y en avoir plus d'une ; les règles pour les combiner sont décrites dans la [RFC4745]). Plus concrètement, si le document de présence envoyé à un observateur est D, et si l'opération de filtrage de confidentialité appliquée à un document de présence x est F(x), alors D DOIT avoir la propriéte que D = F(D). En d'autres termes, d'autres applications de l'opération de filtrage de confidentialité ne résulteraient en aucun autre changement du document de présence, rendant une autre application de l'opération de filtrage un non événement. Un corollaire de cela est que F(F(D)) = D pour tout D.

Les aspects de traitement d'abonnement du document sont appliqués au serveur quand il décide d'accepter ou rejeter l'abonnement.

5. Exigences de mise en œuvre

Les règles définies par le document dans la présente spécification forment un "contrat" de sorts entre un client qui crée ce document et le serveur qui exécute les politiques qu'il contient. Par conséquent, les serveurs de présence qui mettent en œuvre la présente spécification DOIVENT prendre en charge toutes les conditions, actions, et transformations définies dans cette spécification. Si des serveurs devaient mettre en œuvre un sous ensemble de celles-ci, les clients auraient besoin d'un mécanisme pour découvrir quel sous ensemble est pris en charge. Un tel mécanisme n'est pas défini.

Il est RECOMMANDÉ que les clients prennent en charge toutes les actions, transformations, et conditions définies dans la présente spécification. Si un client prend en charge un sous ensemble, il est possible qu'un utilisateur puisse manipuler leurs règles d'autorisation à partir d'un client différent, prenant en charge un sous ensemble différent, et mémoriser ces résultats sur le serveur. Quand l'utilisateur revient au premier client et y voit ses règles d'autorisation de présence, le client peut n'être pas capable de rendre ou manipuler correctement le document restitué du serveur, car il peut contenir des conditions, actions, ou transformations non prises en charge par le client. La seule raison pour laquelle cette exigence normative n'est pas un DOIT est qu'il y a des conditions valides dans lesquelles un utilisateur manipule ses règles d'autorisation de présence à partir d'un seul client, auquel cas ce problème n'apparaît pas.

La présente spécification ne fait pas de recommandations normatives sur le mécanisme utilisé pour transporter les documents d'autorisation de présence des clients à leurs serveurs. Bien que la Section 9 définisse comment utiliser XCAP, XCAP n'est pas une exigence normative de cette spécification.

6. Exemple de document

Le document d'autorisation de présence suivant spécifie les permissions pour l'utilisateur "user@exemple.com". Il est permis à l'observateur d'accéder aux informations de présence (la valeur 'allow' pour <sub-handling>). Il va lui être accordé l'accès aux données de présence de tous les services dont les schéma d'URI de contact sont sip et mailto. Des informations de personne sont aussi fournies. Cependant, comme il n'y a pas de <provide-devices>, aucune information d'appareil ne va être donnée à l'observateur. Dans les informations de service et de personne fournie à l'observateur, l'élément <activities> sera montré, ainsi que l'élement <user-input>. Cependant, tous les attributs "idle-threshold" et "since" seront supprimés

dans l'élément <user-input>. Finalement, l'attribut de présence <foo> va être montré à l'observateur. Tous les autres attributs de présence vont être supprimés.

```
<?xml version="1.0" encoding="UTF-8"?>
<cr:ruleset xmlns="urn:ietf:params:xml:ns:pres-rules"
 xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
 xmlns:cr="urn:ietf:params:xml:ns:common-policy">
 <cr:rule id="a">
  <cr:conditions>
   <cr:identity>
    <cr:one id="sip:user@exemple.com"/>
   </cr:identity>
  </cr:conditions>
  <cr:actions>
   <pr:sub-handling>allow</pr:sub-handling>
  </cr:actions>
  <cr:transformations>
   <pr:provide-services>
     <pr:service-uri-scheme>sip</pr:service-uri-scheme>
     <pr:service-uri-scheme>mailto</pr:service-uri-scheme>
   </pr:provide-services>
   <pr:provide-persons>
     <pr:all-persons/>
   </pr:provide-persons>
   <pr:provide-activities>true</pr:provide-activities>
   <pr:provide-user-input>bare</pr:provide-user-input>
    <pr:provide-unknown-attribute
     ns="urn:vendor-specific:foo-espace de noms"
     name="foo">true</pr:provide-unknown-attribute>
  </cr:transformations>
 </cr:rule>
</cr:ruleset>
```

7. Schéma XML

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema targetNamespace="urn:ietf:params:xml:ns:pres-rules"
   xmlns:xs="http://www.w3.org/2001/XMLSchema"
   xmlns:cr="urn:ietf:params:xml:ns:common-policy"
   xmlns:pr="urn:ietf:params:xml:ns:pres-rules"
   elementFormDefault="qualified" attributeFormDefault="unqualified">
   <xs:import namespace="urn:ietf:params:xml:ns:common-policy"/>
   <xs:simpleType name="booleanPermission">
    <xs:restriction base="xs:boolean"/>
   </xs:simpleType>
   <xs:element name="service-uri-scheme" type="xs:token"/>
   <xs:element name="class" type="xs:token"/>
   <xs:element name="occurrence-id" type="xs:token"/>
   <xs:element name="service-uri" type="xs:anyURI"/>
   <xs:complexType name="provideServicePermission">
    <xs:choice>
     <xs:element name="all-services">
      <xs:complexType/>
     </xs:element>
     <xs:sequence minOccurs="0" maxOccurs="unbounded">
      <xs:choice>
       <xs:element ref="pr:service-uri"/>
       <xs:element ref="pr:service-uri-scheme"/>
       <xs:element ref="pr:occurrence-id"/>
```

```
    <xs:element ref="pr:class"/>
    <xs:any namespace="##other" processContents="lax"/>
   </xs:choice>
  </xs:sequence>
 </xs:choice>
</xs:complexType>
<xs:element name="provide-services"
 type="pr:provideServicePermission"/>
<xs:element name="deviceID" type="xs:anyURI"/>
<xs:complexType name="provideDevicePermission">
 <xs:choice>
  <xs:element name="all-devices">
   <xs:complexType/>
  </xs:element>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
   <xs:choice>
    <xs:element ref="pr:deviceID"/>
    <xs:element ref="pr:occurrence-id"/>
    <xs:element ref="pr:class"/>
    <xs:any namespace="##other" processContents="lax"/>
   </xs:choice>
  </xs:sequence>
 </xs:choice>
</xs:complexType>
<xs:element name="provide-devices"
 type="pr:provideDevicePermission"/>
<xs:complexType name="providePersonPermission">
 <xs:choice>
  <xs:element name="all-persons">
   <xs:complexType/>
  </xs:element>
  <xs:sequence minOccurs="0" maxOccurs="unbounded">
   <xs:choice>
    <xs:element ref="pr:occurrence-id"/>
    <xs:element ref="pr:class"/>
    <xs:any namespace="##other" processContents="lax"/>
   </xs:choice>
  </xs:sequence>
 </xs:choice>
</xs:complexType>
<xs:element name="provide-persons"
 type="pr:providePersonPermission"/>
<xs:element name="provide-activities"
 type="pr:booleanPermission"/>
<xs:element name="provide-class"
 type="pr:booleanPermission"/>
<xs:element name="provide-deviceID"
 type="pr:booleanPermission"/>
<xs:element name="provide-mood"
 type="pr:booleanPermission"/>
<xs:element name="provide-place-est"
 type="pr:booleanPermission"/>
<xs:element name="provide-place-type"
 type="pr:booleanPermission"/>
<xs:element name="provide-privacy"
 type="pr:booleanPermission"/>
<xs:element name="provide-relationship"
 type="pr:booleanPermission"/>
<xs:element name="provide-status-icon"
 type="pr:booleanPermission"/>
<xs:element name="provide-sphere"
```

```
 type="pr:booleanPermission"/>
<xs:element name="provide-time-offset"
 type="pr:booleanPermission"/>
<xs:element name="provide-user-input">
 <xs:simpleType>
  <xs:restriction base="xs:string">
   <xs:enumeration value="false"/>
   <xs:enumeration value="bare"/>
   <xs:enumeration value="thresholds"/>
   <xs:enumeration value="full"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
<xs:element name="provide-note" type="pr:booleanPermission"/>
<xs:element name="sub-handling">
 <xs:simpleType>
  <xs:restriction base="xs:token">
   <xs:enumeration value="block"/>
   <xs:enumeration value="confirm"/>
   <xs:enumeration value="polite-block"/>
   <xs:enumeration value="allow"/>
  </xs:restriction>
 </xs:simpleType>
</xs:element>
<xs:complexType name="unknownBooleanPermission">
 <xs:simpleContent>
  <xs:extension base="pr:booleanPermission">
   <xs:attribute name="name" type="xs:string" use="required"/>
   <xs:attribute name="ns" type="xs:string" use="required"/>
  </xs:extension>
 </xs:simpleContent>
</xs:complexType>
<xs:element name="provide-unknown-attribute"
 type="pr:unknownBooleanPermission"/>
<xs:element name="provide-all-attributes">
 <xs:complexType/>
</xs:element>
</xs:schema>
```

8. Extensibilité de schéma

Il est prévu que de futurs changements de la présente spécification seront accomplis par des extensions qui définiront de nouveaux types de permissions. Ces extensions DOIVENT exister dans un espace de noms différent. De plus, le schéma défini ci-dessus et l'espace de noms pour les éléments définis dans lui NE DOIVENT PAS être altérés par de futures spécifications. Les changements au schéma de base, ou de l'interprétation des éléments au sein de ce schéma, peuvent résulter en des violations de la confidentialité de l'utilisateur dues à une mauvaise interprétation des documents.

Quand des extensions sont faites à l'ensemble de permissions, il devient nécessaire que l'agent qui construit le document de permission (normalement un agent d'utilisateur SIP, mais pas nécessairement) sache quelles permissions sont prises en charge par le serveur. Cela permet à l'agent de savoir comment construire un document qui résulte en le comportement désiré, car des permissions inconnues vont être ignorées par le serveur. Pour traiter cela, quand les documents d'autorisation de présence sont transportés en utilisant XCAP, le document de capacités XCAP mémorisé chez le serveur DEVRAIT contenir les espaces de noms pour les permissions prises en charge par le serveur de présence. De cette façon, un agent peut interroger cette liste avant de construire un document.

9. Usage de XCAP

Cette Section définit les détails nécessaires pour que les clients manipulent les documents d'autorisation de présence à partir d'un serveur en utilisant XCAP.

9.1 Identifiant unique d'application

XCAP exige que les usages d'application définissent un unique identifiant d'usage d'application (AUID, *application usage ID*) dans l'arborescence de l'IETF ou une arborescence de fabricant. La présente spécification définit l'AUID "pres-rules" au sein de l'arborescence IETF, via l'enregistrement IANA de la Section 11.

9.2 Schéma XML

XCAP exige des usages d'application pour définir un schéma pour leurs documents. Le schéma pour les documents d'autorisation de présence est à la Section 7.

9.3 Espace de nom par défaut

XCAP exige des usages d'application pour définir l'espace de noms par défaut pour leurs URI. L'espace de noms par défaut est urn:ietf:params:xml:ns:pres-rules.

9.4 Type MIME

XCAP exige des usages d'application pour définir le type MIME pour les documents qu'elles portent. Les documents d'autorisation de présence héritent du type MIME des documents de politique commune, application/auth-policy+xml.

9.5 Contraintes de validation

Il n'y a pas de contrainte supplémentaire définie par la présente spécification.

9.6 Sémantique des données

La sémantique d'un document d'autorisation de présence est discutée à la Section 3.

9.7 Conventions de désignation

Quand un agent de présence reçoit un abonnement pour un utilisateur foo au sein d'un domaine, il va chercher tous les documents dans http://[xcap root]/pres-rules/users/foo, et utiliser tous les documents trouvés au dessous de ce point pour guider la politique d'autorisation. Si un seul document est utilisé, il DEVRAIT être appelé "index".

9.8 Interdépendances de ressources

Il n'y a pas d'interdépendence de ressources supplémentaire définie par cet usage d'application.

9.9 Politiques d'autorisation

Cet usage d'application ne modifie pas la politique d'autorisation XCAP par défaut, qui est qu'un seul utilisateur peut lire, écrire, ou modifier ses propres documents. Un serveur peut permettre à des utilisateurs privilégiés de modifier les documents qu'ils ne possèdent pas , mais l'établissement et l'indication de telles politiques sortent du domaine d'applicatin du présent document.

10. Considérations sur la sécurité

Les politiques d'autorisation de présence contiennent des informations très sensibles. Elles indiquent quels autres utilisateurs sont "aimés" ou "pas aimés" par un utilisateur. À ce titre, quand ces documents sont transportés sur un réseau, ils DEVRAIENT être chiffrés.

Les modification de ces documents par un attaquant peuvent perturber le service vu par un utilisateur, souvent de façon subtile. Par suite, quand ces documents sont transportés, le transport DEVRAIT assurer l'authenticité et l'intégrité du message.

Dans le cas où XCAP est utilisé pour transférer le document, clients et serveurs DOIVENT mettre en œuvre HTTP sur la sécurité de la couche transport (TLS, *Transport Layer Security*) et l'authentification par résumé HTTP. Les sites DEVRAIENT authentifier les clients en utilisant l'authentification par résumé sur TLS, et les sites DEVRAIENT définir les URI de services racine comme un URI https.

Les documents d'autorisation eux-mêmes existent afin de fournir une fonction de sécurité - confidentialité. Les spécifications de présence SIP [RFC3856] exigent l'usage d'une fonction d'autorisation avant d'accorder les informations de présence, et la présente spécification satisfait à cette exigence. Les documents d'autorisation de présence héritent des propriétés de confidentialité du format de politique commune sur lequel ils se fondent. Ce format a été conçu pour être sûr du point de vue de la confidentialité, ce qui signifie que l'échec du serveur de présence à obtenir ou comprendre un document d'autorisation ne peut jamais révéler plus d'informations que désiré sur l'utilisateur, seulement moins. C'est une conséquence du fait que toutes les permissions sont des attributions positives d'informations, et non des attributions négatives.

Une conséquence de cette conception est que le résultat de la combinaison de plusieurs documents d'autorisation peut n'être pas évident pour les utilisateurs finaux. Par exemple, si un document d'autorisation accorde la permission pour tous les utilisateurs dans le domaine exemple.com de voir leur présence, et si un autre document bloque joe@exemple.com, la combinaison des deux va toujours fournir la présence à joe@exemple.com. Les concepteurs d'interface d'utilisateur sont invités à faire très attention aux résultats de la combinaison de multiples règles.

Un autre souci est celui du cas où un utilisateur règle ses préférences de confidentialité à partir d'un client, charge ses documents d'autorisation de présence sur un serveur, et ensuite les modifie à partir d'un client différent. Si les clients prennent en charge des sous ensembles différents du format de document, les utilisateurs peuvent être embarassés pour savoir quelles informations sont révélées. Les clients qui restituent les documents d'autorisation de présence à partir d'un serveur DEVRAIENT faire apparaître aux utilisateurs les informations sur les règles qu'ils ne comprennent pas, afin que les utilisateurs puissent être certains des règles qui sont en place.

11. Considérations relatives à l'IANA

Plusieurs sujets concernant l'IANA sont associés à la présente spécification.

11.1 Identifiant d'usage d'application XCAP

Ce paragraphe enregistre un identifiant d'usage d'application XCAP (AUID, *Application Usage ID*) en accord avec les procédures pour l'IANA définies dans la [RFC4825].

Nom de l'AUID : pres-rules

Description : les règles de présence sont des documents qui décrivent les permissions qu'une présentité [RFC2778] a accordées aux utilisateurs qui cherchent à observer leur présence.

11.2 Enregistrement de sous espace de nom d'URN

Ce paragraphe enregistre un nouvel espace de noms XML, selon les lignes directrices de la [RFC3688]

URI : l'URI pour cet espace de noms est urn:ietf:params:xml:ns:pres-règles.
Contact de l'enregistrant : groupe de travail IETF SIMPLE (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

XML :

```
DÉBUT
<?xml version="1.0"?>
 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML Basic 1.0//EN"
  "http://www.w3.org/TR/xhtml-basic/xhtml-basic10.dtd">
 <html xmlns="http://www.w3.org/1999/xhtml">
 <head>
   <meta http-equiv="content-type" content="text/html;charset=iso-8859-1"/>
   <title>Presence Rules Namespace</title>
 </head>
 <body>
   <h1>Espace de noms pour les déclarations de permission </h1>
   <h2>urn:ietf:params:xml:ns:pres-rules</h2>
 <p>Voir <a href="http://www.rfc-editor.org/rfc/rfc5025.txt">
 RFC5025</a>.</p>
 </body>
 </html>
FIN
```

11.3 Enregistrements de schéma XML

Ce paragraphe enregistre un schéma XML selon les procédures de la [RFC3688].

URI: urn:ietf:params:xml:schema:pres-règles.
Contact d'enregistrement : IETF, groupe de travail SIMPLE (simple@ietf.org), Jonathan Rosenberg (jdrosen@jdrosen.net).

Le XML pour ce schéma se trouve comme seul contenu de la Section 7.

12. Remerciements

L'auteur tient à remercier Richard Barnes, Jari Urpalainen, Jon Peterson, et Martin Hynar de leurs commentaires.

13. Références

13.1 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)

[RFC2617] J. Franks et autres, "Authentification HTTP : Authentification d'accès de base et par résumé", juin 1999. *(DS.)*

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* 3265, 3853, 4320, 4916, 5393, 6665, 8217, 8760)

[RFC3265] A.B. Roach, "Notification d'événement spécifique du protocole d'initialisation de session (SIP)", juin 2002. (*MàJ par* RFC6446) (*Remplacée par la* RFC6665)

[RFC3323] J. Peterson, "Mécanisme de confidentialité pour le protocole d'initialisation de session (SIP)", novembre 2002.

[RFC3688] M. Mealling, "Registre XML de l'IETF", BCP 81, janvier 2004.

[RFC3857] J. Rosenberg, "Paquetage-gabarit d'événement d'information d'observateur pour le protocole d'initialisation de session (SIP)", août 2004. *(P.S.)*

[RFC3863] H. Sugano et autres, "Format des données d'information de présence (PIDF)", août 2004.

[RFC3966] H. Schulzrinne, "L'URI tel pour les numéros de téléphone", décembre 2004. (*MàJ par* RFC5341) *(P.S.)*

[RFC3987] M. Duerst et M. Suignard, "Identifiant de ressource internationalisé (IRI)", janvier 2005.

[RFC4479] J. Rosenberg, "Modèle de données pour Presence", juillet 2006. *(P.S.)*

[RFC4480] H. Schulzrinne et autres, "RPID : Extensions Rich Presence au format de données d'information Presence (PIDF)", juillet 2006. *(P.S.)*

[RFC4745] H. Schulzrinne et autres, "Politique commune : un format de document pour exprimer les préférences de confidentialité", février 2007. *(P.S.)*

[RFC4825] J. Rosenberg, "Protocole d'accès de configuration (XCAP) du langage de balisage extensible (XML)", mai 2007. *(P.S.)*

13.2 Références pour information

[RFC2778] M. Day, J. Rosenberg et H. Sugano, "Modèle pour Presence et la messagerie instantanée", février 2000.

[RFC3325] C. Jennings, J. Peterson et M. Watson, "Extensions privées au protocole d'initialisation de session (SIP) pour l'assertion d'identité au sein de réseaux de confiance", novembre 2002. *(Information ; MàJ par* RFC8217)

[RFC3856] J. Rosenberg, "Paquetage d'événement Presence pour le protocole d'initialisation de session (SIP)", août 2004.

[RFC4474] J. Peterson et C. Jennings, "Améliorations de la gestion d'identité authentifiée dans le protocole d'initialisation de session (SIP)", août 2006. *(P.S. ; Remplacée par* RFC8224)

Adresse de l'auteur

Jonathan Rosenberg
Cisco
Edison, NJ
US

mél : jdrosen@cisco.com
URI: http://www.jdrosen.net

Déclaration complète de droits de reproduction

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 .