Groupe de travail Réseau
Request for Comments : 3857
Catégorie : En cours de normalisation

J. Rosenberg, dynamicsoft
août 2004
Traduction Claude Brière de L'Isle

Paquetage-gabarit d'événement d'information d'observateur pour le protocole d'initialisation de session (SIP)

Résumé
Le présent document définit le paquetage de gabarit d'information d'observateur pour le cadre d'événement du protocole d'initiation de session (SIP, *Session Initiation Protocol*). Informations d'observateur se réfère à l'ensemble d'utilisateurs abonnés à une ressource particulière au sein d'un paquetage d'événement particulier. Les informations d'observateur changent de façon dynamique lorsque les utilisateurs s'abonnent, se désabonnent, sont approuvés, ou sont rejetés. Un usager peut s'abonner à ces informations, et donc apprendre les changements qu'elles subissent. Ce paquetage d'événement est un paquetage-gabarit parce qu'il peut être appliqué à tout paquetage d'événement, y compris lui-même.

Table des Matières

1. Introduction....1
2. Terminologie....2
3. Scénarios d'utilisation....2
3.1 Autorisation de présence....2
3.2 Alertes de liste noire....3
4. Définition du paquetage....3
4.1 Nom de paquetage d'événement....3
4.2 Paramètres de paquetage d'événement....4
4.3 Corps SUBSCRIBE....4
4.4 Durée d'abonnement....4
4.5 Corps NOTIFY....4
4.6 Traitement des demandes SUBSCRIBE par le notificateur....5
4.7 Génération des demandes NOTIFY par le notificateur....5
4.8 Traitement par l'abonné de demandes NOTIFY....7
4.9 Traitement des demandes fourchées....8
4.10 Taux de notifications....8
4.11 Agents d'état....8
5. Exemple d'utilisation....9
6. Considérations pour la sécurité....10
6.1 Attaques de déni de service....10
6.2 Divulgation d'informations sensibles....11
7. Considérations relatives à l'IANA....11
8. Remerciements....11
9. Références normatives....11
10. Références informatives....12
11. Adresse de l'auteur....12
12. Déclaration complète de droits de reproduction....12

1. Introduction

Le cadre d'événement du protocole d'initiation de session (SIP) est décrit dans la [RFC3265]. Il définit un cadre générique pour l'abonnement et la notification des événements qui se rapportent aux systèmes SIP. Le cadre définit les méthodes

SUBSCRIBE et NOTIFY, et introduit la notion de paquetage. Un paquetage est une application concrète du cadre d'événement à une classe particulière d'événements. Les paquetages ont été définis, par exemple, pour la présence d'utilisateurs [RFC3856].

Le présent document définit un "paquetage-gabarit" au sein du cadre d'événement de SIP. Un paquetage gabarit a toutes les propriétés d'un paquetage d'événement SIP régulier. Cependant, il est toujours associé à quelque autre paquetage d'événement, et peut toujours être appliqué à tout paquetage d'événement, y compris le paquetage gabarit lui-même.

Le gabarit de paquetage défini ici est pour les informations d'observateur, et il est noté avec le jeton "winfo". Pour tout paquetage d'événement, tel que présence, il existe un ensemble (qui peut être un ensemble vide) d'abonnements qui ont été créés ou demandés par les usagers qui essayent de s'assurer de l'état d'une ressource dans ce paquetage. Cet ensemble d'abonnements change au fil du temps lorsque de nouveaux abonnements sont demandés par les usagers, que les vieux abonnements arrivent à expiration, et que des abonnements sont approuvés ou rejetés par les propriétaires de ces ressources. L'ensemble des usagers abonnés à une ressource particulière pour un certain paquetage d'événement, et l'état de leurs abonnements, est appelé les informations d'observateur. Comme cet état est par lui-même dynamique, il est raisonnable de s'y abonner afin d'apprendre ses changements. Le paquetage de gabarit d'événement d'information d'observateur est destiné à faciliter exactement cela – assurant le suivi de l'état des abonnements à une ressource dans un autre paquetage.

Pour noter ce paquetage de gabarit, le nom est construit en ajoutant ".winfo" au nom du paquetage qui est suivi. Par exemple, l'ensemble des gens qui s'abonnent à présence est défini par le paquetage "presence.winfo".

2. Terminologie

Dans ce document, les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" sont à interpréter comme décrit dans le BCP14, [RFC2119] et indiquent les niveaux d'exigence pour les mises en œuvre conformes.

Le présent document traite fondamentalement et de façon récurrente d'abonnement à des abonnements. Donc, le terme "abonnement" peut lui-même prêter à confusion dans le présent document. Pour limiter cette confusion, le terme "abonnement d'informations d'observateur" se réfère à un abonnement à des informations d'observateur, et le terme "abonné aux informations d'observateur" se réfère à un usager qui s'est abonné aux informations d'observateur. Le terme "notification d'informations d'observateur" se réfère à une demande NOTIFIER envoyée au titre d'un abonnement d'informations d'observateur. Lorsque les termes "abonnement", "abonné", et "notification" sont utilisés sans qualification, ils se réfèrent aux abonnements, abonnés et notifications "internes" – ceux qui sont surveillés par les abonnements d'informations d'observateur. On utilise aussi le terme "observateur" pour se référer à un abonné à la ressource "interne". Les informations sur les observateurs sont rapportées par les abonnements d'informations d'observateur.

3. Scénarios d'utilisation

Il y a de nombreuses applications utiles pour le paquetage de gabarit d'informations d'observateur.

3.1 Autorisation de présence

L'application motivante pour ce paquetage de gabarit est l'autorisation de présence. Lorsque l'usager A s'abonne à la présence de l'usager B, l'abonnement doit être autorisé. Fréquemment, cette autorisation doit se faire par une intervention directe de l'usager. Pour que cela arrive, le logiciel de B doit être informé de ce qu'un abonnement à présence a été demandé. Cela est pris en charge par les informations d'observateur. Le logiciel client de B va s'abonner (SUBSCRIBE) aux informations d'observateur pour la présence de B :

```
SUBSCRIBE sip:B@example.com SIP/2.0
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnashds7
From: sip:B@example.com;tag=123s8a
To: sip:B@example.com
Call-ID: 9987@pc34.example.com
Max-Forwards: 70
CSeq: 9887 SUBSCRIBE
Contact: sip:B@pc34.example.com
```

```
Event: presence.winfo
```

La politique du serveur est telle qu'elle permet à B de s'abonner à ses propres informations d'observateur. Ainsi, lorsque A s'abonne à la présence de B, B obtient une notification du changement intervenu dans l'état des informations d'observateur:

```
NOTIFY sip:B@pc34.example.com SIP/2.0
Via: SIP/2.0/UDP server.example.com;branch=z9hG4bKna66g
From: sip:B@example.com;tag=xyz887
To: sip:B@example.com;tag=123s8a
Call-ID: 9987@pc34.example.com
Max-Forwards: 70
CSeq: 1288 NOTIFY
Contact: sip:B@server.example.com
Event: presence.winfo
Content-Type: application/watcherinfo+xml
Content-Length: ...

<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
             version="0" state="full">
    <watcher-list resource="sip:B@example.com" paquetage="presence">
      <watcher id="7768a77s" event="subscribe"
           status="pending">sip:A@example.com</watcher>
    </watcher-list>
  </watcherinfo>
```

Cela indique à B que A s'est abonné, et que l'abonnement est en cours (ce qui signifie qu'il est en attente d'autorisation). Le logiciel de B peut l'alerter sur le fait que cet abonnement est en attente d'autorisation. B peut alors mettre en place une politique pour cet abonnement.

3.2 Alertes de liste noire

Les applications peuvent s'abonner aux informations d'observateur afin de fournir des caractéristiques à valeur ajoutée. Un exemple d'application est "l'alerte de liste noire". Dans ces scénarios, un serveur d'application tient une liste de "mauvais garçons" connus. Un usager, Joe, souscrit au service auprès du fournisseur d'application, vraisemblablement en allant sur une page de la Toile et en entrant dans son URL présence. Le serveur d'application s'abonne aux informations d'observateur pour la présence de Joe. Lorsque quelqu'un tente SUBSCRIBE sur la présence d'utilisateur de Joe, l'application apprend cet abonnement par suite de son abonnement aux informations d'observateur. Elle vérifie si l'URI de l'observateur est dans la base de données des mauvais garçons connus. Si il y est, elle envoie un courriel à Joe pour le lui faire savoir.

Pour que cette application fonctionne, Joe a besoin de s'assurer que l'application est autorisée à s'abonner à ses presence.winfo.

4. Définition du paquetage

La présente section complète les détails nécessaires pour spécifier un paquetage d'événement comme défini au paragraphe 4.4 de la [RFC3265].

4.1 Nom de paquetage d'événement

La [RFC3265] exige que les définitions de paquetage spécifient le nom de leur paquetage ou gabarit de paquetage.

Le nom de ce gabarit de paquetage est "winfo". Il peut être appliqué à tout autre paquetage. Les informations d'observateur pour tout paquetage foo sont notées par le nom "foo.winfo". L'empaquetage récurrent de gabarit est explicitement permis (et utile) de sorte que "foo.winfo.winfo" est un nom de paquetage valide.

4.2 Paramètres de paquetage d'événement

La [RFC3265] exige que les définitions de paquetage et de gabarit de paquetage spécifient tous les paramètres spécifiques du paquetage dans le champ d'en-tête Event.

Aucun paramètre du champ d'en-tête Event spécifique de paquetage n'est défini pour le présent gabarit de paquetage d'événement.

4.3 Corps SUBSCRIBE

La [RFC3265] exige que des définitions de paquetage ou de gabarit de paquetage définissent l'usage, s'il en est, des corps dans les demandes SUBSCRIBE.

Une demande SUBSCRIBE à des informations d'observateur PEUT contenir un corps. Ce corps servirait à filtrer les abonnements watcherinfo. La définition d'un tel corps sort du domaine d'application de la présente spécification. Par exemple, dans le cas de presence, le corps pourrait indiquer que les notifications devraient contenir l'état complet chaque fois que quelque chose change, et que le moment où l'abonnement a été souscrit ne devrait pas être inclus dans les notifications de watcherinfo.

Une demande SUBSCRIBE pour un paquetage d'informations d'observateur PEUT être envoyée sans corps. Cela implique que la politique de filtrage par défaut de l'abonnement à watcherinfo a été demandée. La politique par défaut est :

- Les notifications watcherinfo sont générées chaque fois qu'il y a un changement d'état des informations d'observateur.
- Les notifications watcherinfo déclanchées à partir d'un SUBSCRIBE contiennent l'état complet (la liste de tous les observateurs que l'abonné watcherinfo est autorisé à connaître). Les notifications watcherinfo déclanchées à partir d'un changement d'état d'observateur contiennent seulement des informations sur l'observateur dont l'état a changé.

Bien sûr, le serveur peut appliquer toute politique qu'il veut à l'abonnement.

4.4 Durée d'abonnement

La [RFC3265] exige que les définitions de paquetage définissent une valeur par défaut pour les durées d'abonnement, et proposent des choix raisonnables pour les durées lorsque elles sont explicitement spécifiées.

Les informations d'observateur changent lorsque les usagers s'abonnent à une ressource particulière pour un certain paquetage, ou lorsque leurs abonnements arrivent à expiration. Par suite, l'état des informations d'observateur peut changer de façon très dynamique, selon le nombre d'abonnements pour une certaine ressource dans un certain paquetage. Le taux auquel les abonnements arrivent à expiration dépend de la durée pendant laquelle un usager conserve son abonnement. Normalement, les abonnements watcherinfo vont avoir une durée couvrant la durée de vie des abonnements observés, et vont donc de quelques minutes à plusieurs jours.

Par suite de ces facteurs, il est difficile de définir en gros une valeur par défaut utile pour la durée de vie d'un abonnement à watcherinfo. On choisit arbitrairement une heure. Cependant, les clients DEVRAIENT utiliser un champ d'en-tête Expires pour spécifier leur durée préférée.

4.5 Corps NOTIFY

La [RFC3265] exige que les définitions de paquetage décrivent l'ensemble permis de types de corps dans les demandes NOTIFY, et qu'elles spécifient la valeur par défaut à utiliser lorsque il n'y a pas de champ d'en-tête Accept dans la demande SUBSCRIBE.

Le corps de la notification watcherinfo contient un document d'informations d'observateur. Ce document décrit certains ou tous les observateurs pour une ressource au sein d'un certain paquetage, et l'état de leur abonnement. Tous les abonnés et les notificateurs de watcherinfo DOIVENT prendre en charge le format application/watcherinfo+xml décrit dans la [RFC3858], et DOIVENT faire la liste de son type MIME, application/watcherinfo+xml, dans tout champ d'en-tête Accept présent dans la demande SUBSCRIBE.

D'autres formats d'informations d'observateur pourront être définis à l'avenir. Dans ce cas, les abonnements watcherinfo PEUVENT indiquer leur prise en charge d'autres formats. Cependant, ils DOIVENT toujours prendre en charge, et mettre

dans la liste, application/watcherinfo+xml comme format permis.

Bien sûr, les notifications watcherinfo générées par le serveur DOIVENT être dans un des formats spécifiés dans le champ d'en-tête Accept dans la demande SUBSCRIBE. Si aucun champ d'en-tête Accept n'était présent, les notifications DOIVENT utiliser le format application/watcherinfo+xml décrit dans la [RFC3858].

4.6 Traitement des demandes SUBSCRIBE par le notificateur

La [RFC3265] spécifie que les paquetages devraient définir tout traitement spécifique du paquetage des demandes SUBSCRIBE à un notificateur, en particulier en ce qui concerne l'authentification et l'autorisation.

Les informations d'observateur pour un paquetage particulier contiennent des informations sensibles. Donc, tous les abonnements watcherinfo DEVRAIENT être authentifiés et ensuite autorisés avant approbation. L'authentification PEUT être effectuée en utilisant toute technique disponible par SIP, incluant le résumé, S/MIME, TLS ou autres mécanismes spécifiques du transport [RFC3261]. La politique d'autorisation est à la discrétion de l'administrateur, comme toujours. Cependant, on peut faire quelques recommandations.

Il est RECOMMANDÉ que l'usager A ait la permission de s'abonner à ses propres informations d'observateur pour tout paquetage. C'est vrai par récurrence, de sorte qu'il est RECOMMANDÉ qu'un usager soit capable de s'abonner aux informations d'observateur pour ses informations d'observateur pour tout paquetage.

Il est RECOMMANDÉ que les abonnements watcherinfo pour un paquetage foo pour l'usager A soient permis pour un autre usager B, si B est un abonné autorisé de A dans le paquetage foo. Cependant, il est RECOMMANDÉ que les notifications watcherinfo envoyées à B ne contiennent que l'état du propre abonnement de B. En d'autres termes, il est RECOMMANDÉ qu'un usager soit autorisé à surveiller l'état de son propre abonnement.

Pour éviter des récurrences infinies de politique d'autorisation, il est RECOMMANDÉ que seul l'usager A soit autorisé à s'abonner à foo.winfo.winfo pour l'usager A, pour tout foo. Il est aussi RECOMMANDÉ que par défaut, un serveur n'autorise aucun abonnement à foo.winfo.winfo.winfo ou à aucune autre récurrence plus profonde.

4.7 Génération des demandes NOTIFY par le notificateur

Le cadre d'événement SIP exige que les paquetages spécifient les conditions sous lesquelles les notifications sont envoyées pour ce paquetage, et comment sont construites de telles notifications.

Chaque abonnement watcherinfo est associé à un ensemble d'abonnements "internes" qui sont observés. Cet ensemble est défini par l'URI dans l'URI de demande de la demande SUBSCRIBE à watcherinfo, ainsi que par le paquetage d'événement parent de l'abonnement watcherinfo. Le paquetage d'événement parent est obtenu en retirant le ".winfo" en queue de la valeur du champ d'en-tête Événement de la demande SUBSCRIBE à watcherinfo. Si le champ d'en-tête Événement dans l'abonnement watcherinfo a une valeur de "presence.winfo", le paquetage d'événement parent est "presence". Si le champ d'en-tête Event a une valeur de "presence.winfo.winfo", le paquetage d'événement parent est "presence.winfo". Normalement, l'URI dans l'URI-de-demande du SUBSCRIBE à watcherinfo identifie une adresse d'enregistrement au sein du domaine. Dans ce cas, l'ensemble d'abonnements à observer est tous les abonnements pour le paquetage d'événements parent qui ont été faits à la ressource dans l'URI-de-demande du SUBSCRIBE à watcherinfo. Cependant, l'URI-de-demande peut contenir un URI qui identifie tout ensemble d'abonnements, y compris des abonnements à une plus large collection de ressources. Par exemple, sip:all-resources@example.com pourrait être défini au sein de example.com pour se référer à toutes les ressources. Dans ce cas, un abonnement watcherinfo pour "presence.winfo" à sip:all-resources@example.com demande des notifications chaque fois que change l'état de tout abonnement presence pour toute ressource au sein de example.com. Un notificateur de watcherinfo PEUT générer une notification chaque fois que change l'état d'un des abonnements observés.

Comme un abonnement watcherinfo est fait à une collection d'abonnements, le paquetage d'informations d'observateur a besoin d'un modèle d'état d'abonnement. Ceci se fait en spécifiant un abonnement à un automate à état finis (FSM, *Finite State Machine*) décrit ci-dessous, qui gouverne l'état d'abonnement d'un usager dans tout paquetage. Les notifications watcherinfo PEUVENT être générées sur les transitions dans cet automate à états. Il est important de noter que ce FSM est juste un modèle de la machinerie d'état d'abonnement entretenue par un serveur. Une mise en œuvre transposerait ses propres automates à états en celui-ci d'une manière spécifique.

4.7.1 Automate à état d'abonnement

L'automate à états sous-jacent pour un abonnement est montré à la Figure 1. Il dérive presque entièrement des descriptions de la [RFC3265], mais ajoute la notion d'un état d'attente. Lorsque une demande SUBSCRIBE arrive, l'abonnement FSM est créé dans l'état init. Cet état est transitoire. Le prochain état dépend de si une politique existe pour l'abonnement. Si il y a une politique existante qui détermine que l'abonnement est interdit, il passe immédiatement dans l'état terminé, où le FSM peut être éliminé. Si une politique existante détermine que l'abonnement est autorisé, le FSM passe à l'état actif. Cet état indique que l'abonné va recevoir des notifications. Si, lorsque un abonnement arrive, il n'existe pas de politique d'autorisation, l'abonnement passe à l'état en instance. Dans cet état, le serveur attend une décision d'autorisation. Aucune notification n'est générée sur les changements d'état de présence (un NOTIFY initial aura été livré conformément à la [RFC3265]), mais l'abonnement au FSM est conservé. Si la décision d'autorisation revient positive, l'abonnement est approuvé, et passe à l'état actif. Si l'autorisation est refusée, l'abonnement est rejeté, et le FSM passe à l'état terminé. Il est possible que la décision d'autorisation prenne un temps très long. En fait, aucune décision d'autorisation peut n'arriver avant que l'abonnement lui-même n'arrive à expiration. Si un abonnement en instance subit une fin de temporisation, il passe à l'état en attente. À tout moment, le serveur peut décider de mettre un terme à un abonnement en cours ou en attente parce qu'il a des soucis d'allocation de mémoire et de ressources de CPU à des états d'abonnement non autorisés. Si cela arrive, un événement "abandon" est généré par le serveur, faisant passer l'abonnement à l'état terminé.

L'état d'attente est similaire à celui de en instance, en ce que aucune notification n'est générée. Cependant, si l'abonnement est approuvé ou refusé, le FSM entre dans l'état terminé, et est détruit. De plus, si un autre abonnement est reçu à la même ressource, du même observateur, pour le même paquetage d'événement, les paramètres de paquetage d'événement et le filtre dans le corps de la demande SUBSCRIBE (si il en était une présente initialement) le FSM passe à l'état terminé avec un événement "abandon", et est éliminé. Cette transition se fait parce que, à l'arrivée d'un nouvel abonnement avec des paramètres identiques, il va entrer dans l'état en instance, rendant redondant l'état d'attente pour l'abonnement antérieur. L'objet de l'état en attente est tel qu'un usager peut aller chercher l'état watcherinfo à tout moment, et apprendre tous les abonnements qui sont arrivés précédemment (et qui peuvent arriver à nouveau) qui requièrent une décision d'autorisation. Considérons un exemple. A s'abonne à B. B n'a pas défini de politique sur cet abonnement, et le passe à l'état en instance. B n'est pas "en ligne", de sorte que l'agent logiciel de B ne peut pas être contacté pour approuver l'abonnement. L'abonnement expire. Disons qu'il a été détruit. B se connecte et va chercher ses états watcherinfo state. Il n'y a pas trace de l'abonnement de A, donc aucune décision de politique n'est faite sur les abonnements de A. B se déconnecte. A rafraîchit son abonnement. Une fois de plus, l'abonnement est en instance car aucune politique n'est définie pour lui. Ce processus pourrait continuer indéfiniment. L'état en attente assure que B peut découvrir ces tentatives d'abonnement.

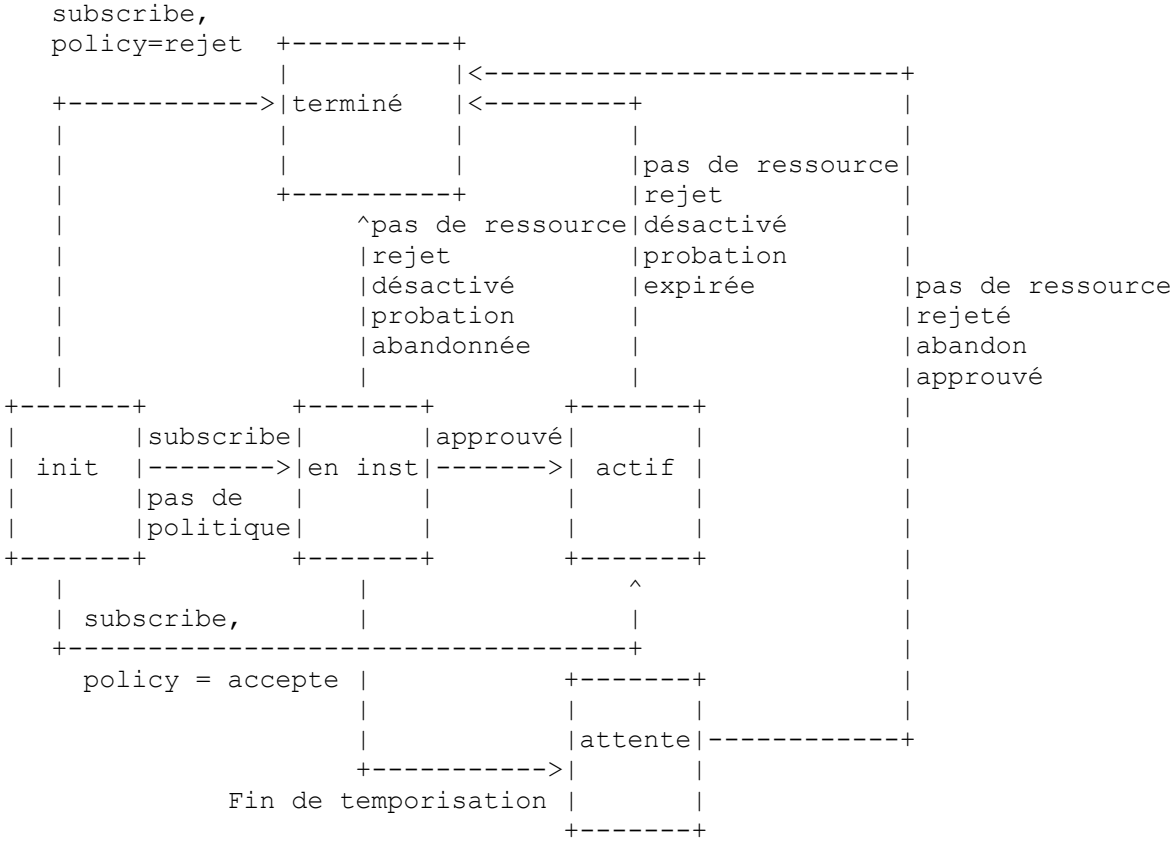

Figure 1 : Automate à états d'abonnement

L'état d'attente est aussi nécessaire pour permettre l'autorisation des tentatives de récupération, qui sont des abonnements qui expirent immédiatement

Bien sûr, une politique peut n'être jamais spécifiée pour l'abonnement. Par suite, le serveur peut générer un événement d'abandon pour passer l'abonnement en attente à l'état terminé. La durée d'attente avant de produire un événement d'abandon dépend du système.

L'événement d'abandon est généré soit dans l'état d'attente, soit dans l'état en instance pour détruire les ressources associées à des abonnements non autorisés. Cet événement est généré lorsque un temporisateur d'abandon arrive à expiration. Ce temporisateur est réglé à une valeur de temporisation lors de l'entrée dans l'un des états en instance ou en attente. Les serveurs doivent faire attention lors du choix de cette valeur. Elle doit être assez grande pour avoir une expérience utile de l'usager ; un usager devrait être capable de se connecter plusieurs jours après et de voir que quelqu'un a essayé de s'abonner à lui. Cependant, allouer un état à des abonnements non autorisés peut être utilisé comme source d'attaques de DoS. Donc, il est RECOMMANDÉ que les serveurs qui conservent l'état pour des abonnements non autorisés ajoutent des politiques qui interdisent à un abonné particulier d'avoir plus qu'un certain nombre d'abonnements en instance ou en attente.

À tout moment, le serveur peut désactiver un abonnement. La désactivation implique que l'abonnement est éliminé sans changer la politique d'autorisation. Cela peut être fait afin de déclancher un rafraîchissement des abonnements pour une clôture en douceur ou une opération de migration de l'abonnement. Un événement en rapport est la probation, où un abonnement est terminé, et il est demandé à l'abonné d'attendre un certain temps avant de réessayer. La signification de ces événements est décrite plus en détails au paragraphe 3.2.4 de la [RFC3265].

Un abonnement peut être terminé à tout moment parce que la ressource associée à cet abonnement n'existe plus. Cela correspond à l'événement 'noresource'.

4.7.2 Application de l'automate à états

Le serveur PEUT générer une notification aux abonnés à watcherinfo sur une transition de l'automate à états. Il dépend de sa politique qu'il le fasse ou non. Cependant, plusieurs lignes directrices sont définies.

Considérons un paquetage d'événement foo. A s'abonne à B pour des événements au sein de ce paquetage. A s'abonne aussi à foo.winfo pour B. Dans ce scénario (où l'abonné à foo.winfo est aussi un abonné à foo pour la même ressource) il est RECOMMANDÉ que A ne reçoive les notifications watcherinfo que sur les changements dans son propre abonnement. Normalement, A va recevoir les notifications sur les changements dans son abonnement à foo par le champ d'en-tête Subscription-State. Cela va souvent combler le besoin d'un abonnement séparé pour foo.winfo. Cependant, si un tel abonnement est effectué par A, les notifications de foo.winfo NE DEVRAIENT PAS rapporter des changements d'état qui ne seraient pas rapportés (à cause de la politique d'autorisation) dans le champ d'en-tête Subscription-State des notifications sur foo.

En règle générale, lorsque un abonné watcherinfo est autorisé à recevoir les notifications watcherinfo sur plus d'un observateur, il est RECOMMANDÉ que les notifications watcherinfo contiennent les informations sur les observateurs qui ont changé d'état (et donc déclanché une notification) au lieu de délivrer l'état en cours à tous les observateurs dans toutes les notifications watcherinfo. Cependant, les notifications watcherinfo déclanchées par suite d'une opération qui est allée les chercher (un SUBSCRIBE avec un Expires de 0) DEVRAIT résulter en ce que l'état complet de tous les observateurs (bien sûr, seuls les observateurs dont il a été autorisé qu'ils soient communiqués à l'abonné watcherinfo) soit présent dans le NOTIFY.

Fréquemment, les états dans l'automate à états d'abonnement vont être transitoires. Par exemple, si un observateur autorisé effectue une opération de collecte, cela va causer la création de l'automate à états, passant de init à active, et ensuite de active à terminée, suivi par une destruction du FSM. Dans un tel cas, les notifications watcherinfo NE DEVRAIENT PAS être envoyées pour des états transitoires. Dans l'exemple précédent, le serveur n'enverrait aucune notification, car tous les états sont transitoires.

4.8 Traitement par l'abonné de demandes NOTIFY

La [RFC3265] prévoit que les paquetages spécifient comment un abonné traite les demandes NOTIFY dans toute façon spécifique du paquetage, et en particulier, comment il utilise les demandes NOTIFY pour construire une vue cohérente de l'état de la ressource souscrite. Normalement, le NOTIFY watcherinfo va seulement contenir des informations sur les observateurs dont l'état a changé. Pour construire une vue cohérente de l'état total de tous les observateurs, un abonné watcherinfo va avoir besoin de combiner les NOTIFY reçus au fil du temps. Ces détails du processus dépendent du format

de document. Voir la [RFC3858] pour les détails du format application/watcherinfo+xml.

4.9 Traitement des demandes fourchées

Le cadre d'événements SIP rend obligatoire que les paquetages indiquent si les demandes SUBSCRIBE fourchées ou non peuvent installer plusieurs abonnements.

Lorsque un usager souhaite obtenir des informations d'observateur pour une ressource pour le paquetage foo, le SUBSCRIBE aux informations d'observateur va devoir atteindre une collection de serveurs qui ont, ensemble, rassemblé toutes les informations sur tous les observateurs sur cette ressource pour le paquetage foo. Si il y a plusieurs serveurs qui traitent les abonnements pour cette ressource pour le paquetage foo (normalement, pour des raisons d'équilibrage de charge) il est très vraisemblable qu'aucun serveur seul n'aura l'ensemble complet des informations d'observateur. Il y a plusieurs solutions dans ce cas. La présente spécification n'en impose, ni n'en exclut, aucune en particulier. Elle s'assure simplement qu'une large gamme de solutions peut être construite.

Une solution est d'utiliser le fourchement. Le système peut être conçu de façon telle qu'arrive un SUBSCRIBE pour les informations d'observateur chez un mandataire spécial qui est au courant des exigences pour les informations d'observateur. Ce mandataire va retransmettre la demande SUBSCRIBE à tous les serveurs qui auraient la possibilité de tenir des abonnements pour cette ressource pour ce paquetage. Chacun de ces serveurs, qu'ils aient ou non des abonnés actuels pour cette ressource, va accepter l'abonnement watcherinfo. Chacun doit accepter parce que tous peuvent finalement recevoir un abonnement pour cette ressource. Les souscripteurs à watcherinfo vont recevoir un certain nombre de demandes NOTIFY watcherinfo, dont chacune établit un dialogue distinct. En agrégeant les informations à travers chaque dialogue, l'abonné watcherinfo peut calculer l'état watcherinfo complet. Dans de nombreux cas, un dialogue particulier peut ne jamais générer de notification watcherinfo ; cela arriverait si les serveurs ne recevaient jamais d'abonnement pour la ressource.

Afin qu'un tel système soit construit de façon interopérable, tous les souscripteurs de watcherinfo DOIVENT être prêts à installer de multiples abonnements par suite d'une multiplicité de messages NOTIFY en réponse à un seul SUBSCRIBE.

Une autre approche pour traiter le problème de la multiplicité des serveurs est d'utiliser les agents d'état. Voir les détails au paragraphe 4.11.

4.10 Taux de notifications

La [RFC3265] rend obligatoire que les paquetages définissent un taux maximum de notifications pour leur paquetage.

Pour des raisons de contrôle d'encombrement, il est important que le taux de notifications ne devienne pas excessif. Par suite, il est RECOMMANDÉ que le serveur ne génère pas les notifications watcherinfo pour un seul abonné watcherinfo à un taux supérieur à une fois toutes les cinq secondes.

4.11 Agents d'état

La [RFC3265] demande que les paquetages considèrent le rôle des agents d'état dans leur conception.

Les agents d'état jouent un rôle important dans le présent paquetage. Comme exposé au paragraphe 4.9, il peut y avoir plusieurs serveurs qui partagent la charge des abonnements pour un paquetage particulier. Un abonnement watcherinfo peut exiger un état d'abonnement éparpillé sur tous ces serveurs. Pour traiter cela, on peut utiliser une "ferme" d'agents d'état. Chacun de ces agents d'état va connaître l'état entier de watcherinfo pour un certain ensemble de ressources. Les moyens par lesquels les agents d'état vont déterminer l'état watcherinfo complet sort du domaine d'application de la présente spécification. Lorsque un abonnement watcherinfo est reçu, il va être acheminé sur un agent d'état qui a l'état watcherinfo complet pour la ressource demandée. Ce serveur va accepter l'abonnement watcherinfo (en supposant bien sûr qu'il y soit autorisé) et générer les notifications watcherinfo lorsque l'état watcherinfo change. L'abonné watcherinfo va seulement avoir un dialogue dans ce cas.

5. Exemple d'utilisation

La présent section expose un exemple de flux d'application et d'appel qui utilise le paquetage watcherinfo.

Dans cet exemple, un usager Joe, sip:joe@example.com fournit la présence à travers le serveur de présence example.com. Joe souscrit à ses propres informations d'observateur, afin d'en savoir plus sur les gens qui s'abonnent à sa présence, de sorte qu'il puisse approuver ou rejeter leur abonnement. Joe envoie la demande SUBSCRIBE suivante :

```
SUBSCRIBE sip:joe@example.com SIP/2.0
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnashds7
From: sip:joe@example.com;tag=123aa9
To: sip:joe@example.com
Call-ID: 9987@pc34.example.com
CSeq: 9887 SUBSCRIBE
Contact: sip:joe@pc34.example.com
Event: presence.winfo
Max-Forwards: 70
```

Le serveur répond par un 401 pour qu'il s'authentifie, et Joe soumet à nouveau le SUBSCRIBE avec des accréditifs (le message n'est pas montré). Le serveur autorise alors l'abonnement, car il permet à Joe de s'abonner à ses propres informations d'observateur pour presence. Il répond par un 200 OK :

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP pc34.example.com;branch=z9hG4bKnashds8
   ;received=192.0.2.8
From: sip:joe@example.com;tag=123aa9
To: sip:joe@example.com;tag=xyzygg
Call-ID: 9987@pc34.example.com
CSeq: 9988 SUBSCRIBE
Contact: sip:server19.example.com
Expires: 3600
Event: presence.winfo
```

Le serveur envoie alors un NOTIFY avec l'état actuel de presence.winfo pour joe@example.com :

```
NOTIFY sip:joe@pc34.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaii
From: sip:joe@example.com;tag=xyzygg
To: sip:joe@example.com;tag=123aa9
Call-ID: 9987@pc34.example.com
CSeq: 1288 NOTIFY
Contact: sip:server19.example.com
Event: presence.winfo
Subscription-State: active
Max-Forwards: 70
Content-Type: application/watcherinfo+xml
Content-Length: ...

<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
        version="0" state="full">
  <watcher-list resource="sip:joe@example.com" package="presence">
    <watcher id="77ajsyy76" event="subscribe"
        status="pending">sip:A@example.com</watcher>
  </watcher-list>
</watcherinfo>
```

Joe répond alors par un 200 OK au NOTIFY :

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaii
  ;received=192.0.2.7
```

```
From: sip:joe@example.com;tag=xyzygg
To: sip:joe@example.com;tag=123aa9
Call-ID: 9987@pc34.example.com
CSeq: 1288 NOTIFY
```

Le NOTIFY dit à Joe que l'usager A a actuellement un abonnement en cours. Joe autorise alors l'abonnement de A par un moyen quelconque. Cela cause un changement dans l'état de l'abonnement (qui passe de l'état en cours à actif) et la livraison d'une autre notification :

```
NOTIFY sip:joe@pc34.example.com SIP/2.0
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaij
From: sip:joe@example.com;tag=xyzygg
To: sip:joe@example.com;tag=123aa9
Call-ID: 9987@pc34.example.com
CSeq: 1289 NOTIFY
Contact: sip:server19.example.com
Event: presence.winfo
Subscription-State: active
Max-Forwards: 70
Content-Type: application/watcherinfo+xml
Content-Length: ...

<?xml version="1.0"?>
<watcherinfo xmlns="urn:ietf:params:xml:ns:watcherinfo"
        version="1" state="partial">
 <watcher-list resource="sip:joe@example.com" package="presence">
  <watcher id="77ajsyy76" event="approved"
        status="active">sip:A@example.com</watcher>
 </watcher-list>
</watcherinfo>
```

B répond alors par un 200 OK au NOTIFY:

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP server19.example.com;branch=z9hG4bKnasaij
   ;received=192.0.2.7
From: sip:joe@example.com;tag=xyzygg
To: sip:joe@example.com;tag=123aa9
Call-ID: 9987@pc34.example.com
CSeq: 1289 NOTIFY
```

6. Considérations pour la sécurité

6.1 Attaques de déni de service

Les informations d'observateur génèrent des notifications sur les changements de l'état des observateurs pour une certaine ressource. Il est possible qu'une seule ressource ait plusieurs observateurs, d'où résulte la possibilité d'un gros volume de notifications. Cela fait de l'abonnement aux informations d'observateur un outil potentiel d'attaques de déni de service. On peut empêcher cela par une combinaison de politiques d'autorisation intelligentes et de bons principes de fonctionnement.

D'abord, lorsque une ressource a beaucoup d'observateurs, l'abonnement aux informations d'observateur pour cette ressource ne devrait être permis qu'à partir d'entités explicitement autorisées, dont l'identité a été proprement authentifiée. Cela empêche qu'un flux de NOTIFY d'informations d'observateur soit généré à partir d'abonnements faits par un attaquant.

Même lorsque l'abonnement aux informations d'observateur a été proprement authentifié, il y a encore un potentiel d'attaques. Par exemple, considérons un utilisateur valide, T, qui est la cible d'une attaque. T a souscrit à ses propres informations d'observateur. L'attaquant génère un grand nombre d'abonnements (pas des abonnement watcherinfos). Si le serveur crée un état d'abonnement pour des abonnements non authentifiés, et rapporte ces changements dans les notifications watcherinfo, l'usager T va recevoir un flot de notifications watcherinfo. En fait, si le serveur génère une notification watcherinfo lorsque l'abonnement est créé, et un autre lorsque il est terminé, il y aura une amplification d'un

facteur deux. L'amplification sera en fait substantielle si le serveur génère un état complet dans chaque notification watcherinfo. Bien sûr, la quantité de données envoyées à T sera le carré des données générées par l'attaquant ! Chacun des N abonnements générés par l'attaquant va résulter en l'envoi d'un NOTIFY watcherinfo à T, dont chacun va rapporter jusqu'à N observateurs. Pour éviter cela, les serveurs ne devraient jamais générer d'état d'abonnement pour des demandes SUBSCRIBE non authentifiées, et ne devraient jamais non plus générer de notifications pour elles.

6.2 Divulgation d'informations sensibles

Les informations d'observateur indiquent quels utilisateurs sont intéressés par une certaine ressource. Selon le paquetage et la ressource, cela peut être une information très sensible. Par exemple, dans le cas de presence, les informations d'observateur pour un certain usager représentent les amis, la famille, et les relations d'affaire de cette personne. Cette information peut être utilisée pour divers objets malveillants.

Une façon dont ces informations peuvent être révélées est l'espionnage. Un attaquant peut observer les notifications watcherinfo, et apprendre ces informations. Pour l'empêcher, les observateurs PEUVENT utiliser le schéma d'URI sips lorsque ils souscrivent à une ressource watcherinfo. Les notificateurs pour watcherinfo DOIVENT prendre en charge TLS et sips comme si ils étaient un mandataire (voir le paragraphe 26.3.1 de la RFC 3261).

Le chiffrement SIP, qui utilise S/MIME, PEUT être utilisé de bout en bout pour la transmission des demandes SUBSCRIBE aussi bien que NOTIFY.

Une autre façon dont ces informations peuvent être révélées est par des abonnements falsifiés. Ces attaques peuvent être empêchées en authentifiant et en autorisant tous les abonnements watcherinfo. Pour que le notificateur authentifie le souscripteur, il PEUT utiliser le résumé HTTP (Section 22 de la RFC 3261). Par suite, tous les observateurs DOIVENT prendre en charge HTTP Digest. Cette exigence est cependant redondante, car tous les agents d'utilisateur SIP sont obligés de le prendre en charge par la RFC 3261.

7. Considérations relatives à l'IANA

La présente spécification enregistre un paquetage de gabarit d'événement comme spécifié au paragraphe 6.2 de la [RFC3265].

Nom du paquetage : winfo
Gabarit de paquetage : oui
Spécification publiée : RFC 3857
Personne à contacter : Jonathan Rosenberg, jdrosen@jdrosen.net.

8. Remerciements

Les auteurs tiennent à remercier Adam Roach, Allison Mankin et Brian Stucker pour leurs commentaires détaillés.

9. Références normatives

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

[RFC3261] J. Rosenberg et autres, "SIP : Protocole d'initialisation de session", juin 2002. (*Mise à jour par* RFC3265, RFC3853, RFC4320, RFC4916, RFC5393, RFC6665))

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

[RFC3858] J. Rosenberg, "Format fondé sur le langage de balisage extensible (XML) pour les informations d'observateur", août 2004. *(P.S.)*

10. Références informatives

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

11. Adresse de l'auteur

Jonathan Rosenberg
dynamicsoft
600 Lanidex Plaza
Parsippany, NJ 07054
USA
mél : jdrosen@dynamicsoft.com

12. Déclaration complète de droits de reproduction

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