Network Working Group
Request for Comments: 5362
Category: Standards Track

G. Camarillo
Ericsson

October 2008

The Session Initiation Protocol (SIP) Pending Additions Event Package

Status of This Memo

Abstract

This document defines the SIP Pending Additions event package. This event package is used by SIP relays to inform user agents about the consent-related status of the entries to be added to a resource list.

Table of Contents

1. Introduction

The framework for consent-based communications in SIP [RFC5360] identifies the need for users manipulating the translation logic at a relay (e.g., adding a new recipient) to be informed about the consent-related status of the recipients of a given translation. That is, the user manipulating the translation logic needs to know which recipients have given the relay permission to send them SIP requests.

This document defines a SIP event package whereby user agents can subscribe to the consent-related state of the resources that are being added to a resource list that defines a translation.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Relay: Any SIP server, be it a proxy, B2BUA (Back-to-Back User Agent), or some hybrid, that receives a request, translates its Request-URI into one or more next-hop URIs (i.e., recipient URIs), and delivers the request to those URIs.

3. Overview of Operation

A user agent subscribes to a relay using the Pending Additions event package. NOTIFY requests within this event package can carry an XML document in the "application/resource-lists+xml" format [RFC4826] or in the "application/resource-lists-diff+xml" format, which is based on XML patch operations [RFC5261].

A document in the "application/resource-lists+xml" format provides the user agent with the whole list of resources being added to a resource list along with the consent-related status of those resources. A document in the "application/resource-lists-diff+xml" format provides the user agent with the changes the list of resources being added has experimented with since the last notification sent to the user agent.

4. XML Schema Definition

This section defines the <consent-status> element, which provides consent-related information about a resource to be added to a relay's translation logic.

A consent-status document is an XML document that MUST be well-formed and SHOULD be valid. Consent-status documents MUST be based on XML 1.0 and MUST be encoded using UTF-8. This specification makes use of XML namespaces for identifying consent-status documents. The namespace URI for elements defined for this purpose is a URN, using the namespace identifier 'ietf'. This URN is:

urn:ietf:params:xml:ns:consent-status

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema targetNamespace="urn:ietf:params:xml:ns:consent-status"
  elementFormDefault="qualified"
  attributeFormDefault="unqualified"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="urn:ietf:params:xml:ns:consent-status">
   <xs:element name="consent-status">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:enumeration value="pending"/>
          <xs:enumeration value="waiting"/>
          <xs:enumeration value="error"/>
          <xs:enumeration value="denied"/>
          <xs:enumeration value="granted"/>
        </xs:restriction>
      </xs:simpleType>
   </xs:element>
</xs:schema>
```

The <consent-status> element can take on the following values:

Pending: the relay has received a request to add a resource to its translation logic and will ask for permission to do so.

Waiting: the relay has requested permission to add the resource to its translation logic but has not gotten any answer from the resource yet.

Error: the relay has requested permission to add the resource to its translation logic and has received an error response (e.g., a SIP error response to the MESSAGE request sent to request permission). That is, the permission document requesting permission could not be delivered to the resource.

Denied: the resource has denied the relay permission to add the resource to the relay's translation logic.

Granted: the resource has granted the relay permission to add the resource to the relay's translation logic.

Section 5.1.11 contains an example of an "application/resource-lists+xml" document that carries consent-related state information using <consent-status> elements.

5. Pending Additions Event Package Definition

This section provides the details for defining a SIP [RFC3261] event notification package, as specified by [RFC3265]. Support for this section (i.e., Section 5) is REQUIRED for implementations of this specification. Support for partial notifications is optional, but if a subscriber signals support for partial notifications, Section 6 MUST be implemented.

5.1. Event Package Name

The name of this event package is "consent-pending-additions". This package name is carried in the Event and Allow-Events header, as defined in [RFC3265].

5.1.1. Event Package Parameters

This package does not define any event package parameters.

5.1.2. SUBSCRIBE Bodies

A SUBSCRIBE for Pending Additions events MAY contain a body. This body would serve the purpose of filtering the subscription. Filter documents are not specified in this document and, at the time of writing, they are expected to be the subject of future standardization activity.

A SUBSCRIBE for the Pending Additions event package MAY be sent without a body. This implies that the default session policy filtering policy has been requested. The default policy is that notifications are generated every time there is any change in the state of a resource in the list.

5.1.3. Subscription Duration

The default expiration time for a subscription is one hour (3600 seconds).

5.1.4. NOTIFY Bodies

In this event package, the body of the notifications contains a resource list document. This document describes the resources being added as recipients to a translation operation. All subscribers and notifiers MUST support the "application/resource-lists+xml" data

format [RFC4826] and its extension to carry consent-related state information, which is specified in Section 4. The SUBSCRIBE request MAY contain an Accept header field. If no such header field is present, it has a default value of "application/resource-lists+xml". If the header field is present, it MUST include "application/resource-lists+xml", and MAY include any other types capable of representing consent-related state.

Additionally, all subscribers and notifiers SHOULD support the "application/resource-lists-diff+xml" format. Section 6 discusses the usage of the Pending Additions event package with this format.

5.1.5. Notifier Processing of SUBSCRIBE Requests

The state of the resources to be added to a relay's translation logic can reveal sensitive information. Therefore, all subscriptions SHOULD be authenticated and then authorized before approval. Authorization policy is at the discretion of the administrator.

5.1.6. Notifier Generation of NOTIFY Requests

A notifier for the Pending Additions event package SHOULD include the <consent-status> element, which is defined in Section 4. The <consent-status> element MUST be positioned as an instance of the <any> element within the <entry> element.

Notifications SHOULD be generated for the Pending Additions package whenever there is a change in the consent-related state of a resource. When a resource moves to the error, denied, or granted states, and once a NOTIFY request is sent, the resource is removed from further notifications.

5.1.7. Subscriber Processing of NOTIFY Requests

As stated in Section 3, a document in the "application/resource-lists+xml" format provides the subscriber with the whole list of resources being added to a resource list along with the consent-related status of those resources. On receiving a NOTIFY request with such a document, the subscriber SHOULD update its local information about the resources being added to the resource list with the information in the document. NOTIFY requests contain full state. The subscriber does not need to perform any type of information aggregation. Section 6 discusses the use of the Pending Additions event package with partial notifications.

5.1.8. Handling of Forked Requests

The state of a given resource list is normally handled by a server and stored in a repository. Therefore, there is usually a single place where the resource-list state is resident. This implies that a subscription for this information is readily handled by a single element with access to this repository. There is, therefore, no compelling need for a subscription to pending additions information to fork. As a result, a subscriber MUST NOT create multiple dialogs as a result of a single subscription request. The required processing to guarantee that only a single dialog is established is described in Section 4.4.9 of [RFC3265].

5.1.9. Rate of Notifications

For reasons of congestion control, it is important that the rate of notifications not become excessive. As a result, it is RECOMMENDED that the server does not generate notifications for a single subscriber at a rate faster than once every 5 seconds.

5.1.10. State Agents

State agents have no role in the handling of this package.

5.1.11. Example

The following is an example of an "application/resource-lists+xml" document that carries consent-related state information using <consent-status> elements:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
 xmlns:cs="urn:ietf:params:xml:ns:consent-status">
 <list>
  <entry uri="sip:bill@example.com">
   <display-name>Bill Doe</display-name>
   <cs:consent-status>pending</cs:consent-status>
  </entry>
  <entry uri="sip:joe@example.com">
   <display-name>Joe Smith</display-name>
   <cs:consent-status>pending</cs:consent-status>
  </entry>
  <entry uri="sip:nancy@example.com">
   <display-name>Nancy Gross</display-name>
   <cs:consent-status>granted</cs:consent-status>
  </entry>
 </list>
</resource-lists>
```

6. Partial Notifications

The lists of resources reported by this event package may contain many resources. When the "application/resource-lists+xml" format is used and there is a change in the consent-related status of a resource, the server generates a notification with the whole list. Generating large notifications to report small changes does not meet the efficiency requirements of some bandwidth-constrained environments. The partial notifications mechanism specified in this section is a more efficient way to report changes in the status of resources.

Subscribers signal support for partial notifications by including the "application/resource-lists-diff+xml" format in the Accept header field of the SUBSCRIBE requests they generate. If a client subscribing to the Pending Additions event package generates an Accept header field that includes the MIME type "application/resource-lists-diff+xml", the server has the option of returning documents in this format (instead of in the "application/resource-list+xml" format).

6.1. Generation of Partial Notifications

Once a subscription is accepted and installed, the server MUST deliver full state in its first notification. To report full state, the server uses the regular format for resource lists. Consequently, the server MUST set the Content-Type header field to the value 'application/resource-lists+xml'.

In order to deliver a partial notification, the server MUST set the Content-Type header field to the value 'application/resource-lists-diff+xml'. When the server generates a partial notification, the server SHOULD only include the information that has changed since the previous notification. It is up to the server's local policy to determine what is considered as a change to the previous state.

The server MUST construct partial notifications according to the following logic: all information that has been added to the document is listed inside <add> elements, all information that has been removed from the document is listed inside <remove> elements, and all information that has been changed is listed under <replace> elements.

The server MUST NOT send a new NOTIFY request with a partial notification until it has received a final response from the subscriber for the previous one or the previous NOTIFY request has timed out.

When the server receives a SUBSCRIBE request (refresh or termination) within the associated subscription, it SHOULD send a NOTIFY request containing the full document using the 'application/resource-lists+xml' content type.

If the server has used a content type other than 'application/resource-lists+xml' in notifications within the existing subscription and changes to deliver partial notifications, the server MUST deliver full state using the 'application/resource-lists+xml' content type before generating its first partial notification.

6.2. Processing of Partial Notifications

When a subscriber receives the first notification containing full state in a 'application/resource-lists+xml' MIME body, the subscriber MUST store the received full document as its local copy.

When the subscriber receives a subsequent notification, the subscriber MUST modify its locally stored information according to the following logic:

- If the notification carries an %'application/resource-lists+xml' document, the subscriber MUST replace its local copy of the document with the document received in notification.

- If the notification carries an 'application/resource-lists-diff+xml' document, the subscriber MUST apply the changes indicated in the received 'application/resource-lists-diff+xml' document to its local copy of the full document.

If a subscriber encounters a processing error while processing an 'application/resource-lists-diff+xml' encoded document, the subscriber SHOULD renew its subscription. A subscriber can fall back to normal operations by not including the 'application/resource-list-diff+xml' format in a new SUBSCRIBE request.

If the server changes the content type used in notifications within the existing subscription, the subscriber MUST discard all the previously received information and process the new content as specified for that content type.

6.3. XML Schema for Partial Notifications

This is the XML schema for the "application/resource-lists-diff+xml" data format. The "urn:ietf:params:xml:schema:xml-patch-ops" schema is defined in [RFC5261].

```
<?xml version="1.0" encoding="UTF-8"?>
  <xs:schema
         targetNamespace="urn:ietf:params:xml:ns:resource-lists"
         xmlns="urn:ietf:params:xml:ns:resource-lists"
         xmlns:xs="http://www.w3.org/2001/XMLSchema"
         elementFormDefault="qualified">

    <!-- include patch-ops type definitions -->
     <xs:include
          schemaLocation="urn:ietf:params:xml:schema:patch-ops"/>

    <!-- partial updates -->
   <xs:element name="resource-lists-diff">
    <xs:sequence minOccurs="0" maxOccurs="unbounded">
     <xs:choice>
      <xs:element name="add">
       <xs:complexType mixed="true">
        <xs:complexContent>
         <xs:extension base="add">
          <xs:anyAttribute processContents="lax"/>
         </xs:extension>
        </xs:complexContent>
       </xs:complexType>
      </xs:element>
      <xs:element name="remove">
       <xs:complexType>
        <xs:complexContent>
         <xs:extension base="remove">
          <xs:anyAttribute processContents="lax"/>
         </xs:extension>
        </xs:complexContent>
       </xs:complexType>
      </xs:element>
      <xs:element name="replace">
       <xs:complexType mixed="true">
        <xs:complexContent>
         <xs:extension base="replace">
          <xs:anyAttribute processContents="lax"/>
         </xs:extension>
        </xs:complexContent>
       </xs:complexType>
      </xs:element>
      <xs:any namespace="##other" processContents="lax"/>
     </xs:choice>
    </xs:sequence>
    <xs:anyAttribute processContents="lax"/>
   </xs:element>
  </xs:schema>
```

6.4. Examples

Section 5.1.11 contains an example of an 'application/resource-lists+xml' document, which carries full state. The following is an 'application/resource-lists-diff+xml' partial update document:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists-diff xmlns="urn:ietf:params:xml:ns:resource-lists"
 xmlns:cs="urn:ietf:params:xml:ns:consent-status">

<replace
sel="*/list/entry[@uri='sip:bill@example.com']/cs:consent-status/text()"
>granted</replace>

</resource-lists-diff>
```

The following is the resulting 'application/resource-lists+xml' document after applying the partial update:

```
<?xml version="1.0" encoding="UTF-8"?>
<resource-lists xmlns="urn:ietf:params:xml:ns:resource-lists"
 xmlns:cs="urn:ietf:params:xml:ns:consent-status">
 <list>
  <entry uri="sip:bill@example.com">
   <display-name>Bill Doe</display-name>
   <cs:consent-status>granted</cs:consent-status>
  </entry>
  <entry uri="sip:joe@example.com">
   <display-name>Joe Smith</display-name>
   <cs:consent-status>pending</cs:consent-status>
  </entry>
  <entry uri="sip:nancy@example.com">
   <display-name>Nancy Gross</display-name>
   <cs:consent-status>granted</cs:consent-status>
  </entry>
 </list>
</resource-lists>
```

7. IANA Considerations

There are five IANA considerations associated with this specification.

7.1. SIP Event Package Registration

This specification registers a SIP event package per the procedures in [RFC3265].

Package name: consent-pending-additions

Type: package

Contact: Gonzalo Camarillo <Gonzalo.Camarillo@ericsson.com>

Published Specification: RFC 5362.

7.2. URN Sub-Namespace Registration: consent-status

This section registers a new XML namespace per the procedures in [RFC3688].

URI: The URI for this namespace is urn:ietf:params:xml:ns:consent-status

Registrant Contact: IETF SIPPING working group <sipping@ietf.org>, Gonzalo Camarillo <Gonzalo.Camarillo@ericsson.com>

XML:

```
<?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>Pending Additions Extension Namespace</title>
</head>
<body>
  <h1>Namespace for Consent-related Status Information Extension</h1>
  <h2>urn:ietf:params:xml:ns:consent-status</h2>
  <p>See <a href="http://www.rfc-editor.org/rfc/rfc5362.txt">RFC 5362
    </a>.</p>
 </body>
</html>
```

7.3. XML Schema Registration: consent-status

This section registers an XML schema per the procedures in [RFC3688].

URI: urn:ietf:params:xml:schema:consent-status

Registrant Contact: IETF SIPPING working group <sipping@ietf.org>, Gonzalo Camarillo <Gonzalo.Camarillo@ericsson.com>

The XML for this schema can be found in Section 4.

7.4. XML Schema Registration: resource-lists

This section registers an XML schema per the procedures in [RFC3688]. This XML schema is an extension to the XML schema (whose ID is resource-list) defined in [RFC4826]. The IANA has added a row in the XML schema registry with the following values:

ID: resource-list-diff
URI: urn:ietf:params:xml:schema:resource-lists-diff
Filename: resource-lists-diff
Reference [RFC5362]

Registrant Contact: IETF SIPPING working group <sipping@ietf.org>, Gonzalo Camarillo <Gonzalo.Camarillo@ericsson.com>

The XML for this schema can be found in Section 6.3.

7.5. MIME Type Registration: application/resource-lists-diff+xml

This section registers the 'application/resource-lists-diff+xml' MIME type.

MIME media type name: application
MIME subtype name: resource-lists-diff+xml
Mandatory parameters: none
Optional parameters: Same as charset parameter application/xml as specified in [RFC3023].
Encoding considerations: Same as encoding considerations of application/xml as specified in [RFC3023].
Security considerations: See Section 10 of [RFC3023] and Section 7 of [RFC4826].

Interoperability considerations: none
Published specification: RFC 5362
Applications that use this media type: This document type has been defined to support partial notifications in subscriptions to resource lists.

Additional Information:

Magic number: none
File extension: .rld
Macintosh file type code: "TEXT"
Personal and email address for further information: Gonzalo Camarillo <Gonzalo.Camarillo@ericsson.com>
Intended usage: COMMON
Author/Change controller: The IETF

8. Security Considerations

"A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)" [RFC5360] discusses security-related issues that are related to this specification.

Subscriptions to the Pending Additions event package can reveal sensitive information. For this reason, it is RECOMMENDED that relays use strong means for authentication and information confidentiality. Additionally, attackers may attempt to modify the contents of the notifications sent by a relay to its subscribers. Consequently, it is RECOMMENDED that relays use a strong means for information integrity protection.

It is RECOMMENDED that relays authenticate subscribers using the normal SIP authentication mechanisms, such as Digest, as defined in [RFC3261].

The mechanism used for conveying information to subscribers SHOULD ensure the integrity and confidentially of the information. In order to achieve these, an end-to-end SIP encryption mechanism, such as S/MIME, as described in [RFC3261], SHOULD be used.

If strong end-to-end security means (such as above) is not available, it is RECOMMENDED that hop-by-hop security based on TLS and SIPS URIs, as described in [RFC3261], is used.

9. Acknowledgments

Jonathan Rosenberg provided useful ideas on this document. Ben Campbell and Mary Barnes performed a thorough review of this document. Jari Urpalainen helped improve the partial notifications mechanism.

10. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3265] Roach, A., "Session Initiation Protocol (SIP)-Specific Event Notification", RFC 3265, June 2002.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, January 2004.

[RFC4826] Rosenberg, J., "Extensible Markup Language (XML) Formats for Representing Resource Lists", RFC 4826, May 2007.

[RFC5261] Urpalainen, J., "An Extensible Markup Language (XML) Patch Operations Framework Utilizing XML Path Language (XPath) Selectors", RFC 5261, September 2008.

[RFC5360] Rosenberg, J., Camarillo, G., and D. Willis, "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", RFC 5360, October 2008.

Author's Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property