SIPPING
Internet-Draft
Intended status: Standards Track
Expires: May 16, 2008

G. Camarillo
Ericsson

November 13, 2007

The Session Initiation Protocol (SIP) Pending Additions Event Package draft-ietf-sipping-pending-additions-03.txt

Status of this Memo

Copyright Notice

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 [I-D.ietf-sip-consent-framework] 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/xcap-diff+xml" format [I-D.ietf-simple-xcap-diff].

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/xcap-diff+xml" format informs the user agent that the document that describes the resources being added to the resource list has changed. The user agent can then download the document in the "application/resource-lists+xml" format from the relay using XCAP [RFC4825].

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 send 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.

5. Pending Additions Event Package Definition

This section provides the details for defining a SIP [RFC3261] event notification package, as specified by [RFC3265].

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. The definition of such a body is outside the scope of this specification.

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/xcap-diff+xml" format [I-D.ietf-simple-xcap-diff]. 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

NOTIFY requests contain full state. The subscriber does not need to perform any type of information aggregation.

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:xsi="http://www.w3.org/2001/XMLSchema-instance"
 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. Usage of the Pending Additions Event Package with the XCAP Diff Format

As discussed in Section 5.1.4, if a client subscribing to the Pending Additions event package generates an Accept header field that includes the MIME type "application/xcap-diff+xml", the relay has the option of returning documents in this format (instead of in the 'application/resource-list+xml' format).

Upon initial subscription, the relay does not know which instance of the resource list document for the user (where each instance is identified by an etag) the client currently possesses, if any. Indeed, upon startup, the client will not have any documents.

The initial NOTIFY request in this case MUST include a <document> element for the resource list. The "previous-etag" attribute MUST be absent, and the "new-etag" attribute MUST be present and contain the entity tag for the current version of the document. An XCAP diff document structured this way is called a "reference" XCAP diff document. It establishes the baseline etag and document URI for the document covered by the subscription.

Upon receipt of this document, the client can determine whether its local instance document, if any, matches the etag in the XCAP diff document. If they do not match, the client SHOULD perform a conditional GET for each document. The document URI is constructed by appending the XCAP root in the "xcap-root" attribute of the <xcap-diff> element to the escape coded "doc-selector" from the <document> element. The request is made conditional by including an If-Match header field, with the value of the etag from the <document> element. So long as the documents have not changed between the NOTIFY and the GET, the client will obtain the reference version that the server will use for subsequent notifications.

If the conditional GET should fail, the client SHOULD generate a SUBSCRIBE refresh request to trigger a new NOTIFY. The server will always generate a "reference" XML diff document on receipt of a SUBSCRIBE refresh. This establishes a new baseline etag, and the client can then attempt to do another fetch.

Once the client has obtained the version of the document identified in the reference XML diff, it can process NOTIFY requests on that subscription. To process the NOTIFY requests, it makes sure that its current version matches the version in the "previous-etag" attribute of the <document> element. If not, the client can then fetch the updated document from the server. If they do match, the client has the most current version.

7. IANA Considerations

There are three 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 XXXX. (Note to the RFC Editor: Please replace XXXX with the RFC Number of this specification.)

7.2. URN Sub-Namespace Registration

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="[URL of published RFC]">RFCXXXX [[NOTE TO
RFC-EDITOR/IANA: Please replace XXXX with the RFC Number of
this specification]]</a>.</p>
 </body>
</html>
```

7.3. XML Schema Registration

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.

8. Security Considerations

The Framework for Consent-based Communications in the Session Initiation Protocol (SIP) [I-D.ietf-sip-consent-framework] discusses security-related issues that are related to this specification.

Subscriptions to the Pending Additions even 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 clients. 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 clients 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. Acknowledgements

Jonathan Rosenberg provided useful ideas on this document. Ben Campbell and Mary Barnes performed a thorough review of this document.

10. Normative References

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

[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.

[RFC4825] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", RFC 4825, May 2007.

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

[I-D.ietf-simple-xcap-diff] Rosenberg, J., "An Extensible Markup Language (XML) Document Format for Indicating A Change in XML Configuration Access Protocol (XCAP) Resources", draft-ietf-simple-xcap-diff-05 (work in progress), March 2007.

[I-D.ietf-sip-consent-framework]
Rosenberg, J., "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", draft-ietf-sip-consent-framework-01 (work in progress), November 2006.

Author's Address

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).