                Explicit Subscriptions for the REFER Method

Abstract

   The Session Initiation Protocol (SIP) REFER request, as defined by
   RFC 3515, triggers an implicit SIP-Specific Event Notification
   framework subscription. Conflating the start of the subscription
   with handling the REFER request makes negotiating SUBSCRIBE
   extensions impossible and complicates avoiding SIP dialog sharing.
   This document defines extensions to REFER that remove the implicit
   subscription and, if desired, replace it with an explicit one.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc7614.

Table of Contents

1. Introduction

   REFER, as defined by [RFC3515], triggers an implicit SIP-Specific
   Event Framework subscription. Sending a REFER within a dialog
   established by an INVITE results in dialog reuse and the associated
   problems described in [RFC5057]. The SIP-Specific Event Notification
   framework definition [RFC6665] disallows such dialog reuse. Call
   transfer, as defined in [RFC5589], thus requires sending a REFER
   request on a new dialog, associating it with an existing dialog using
   the 'Target-Dialog' mechanism defined in [RFC4538].

   Because there is no explicit SUBSCRIBE request, the tools for
   negotiating subscription details are unavailable for REFER
   subscriptions. This includes negotiating subscription duration and
   providing information through Event header field parameters. The use
   of the SIP 'Supported' and 'Require' extension mechanisms [RFC3261]
   is complicated by the implicit subscription. It is unclear whether
   or not the extension applies to handling the REFER request itself, to
   the messages in the subscription created by the REFER, or to both.
   Avoiding this confusion requires careful specification in each
   extension. Existing extensions do not provide this clarity.

   This document defines two mechanisms that remove the implicit
   subscription, one of which replaces it with an explicit one. The
   benefits of doing so include:

   o Allowing REFER to be used within INVITE-created dialogs without
      creating dialog reuse.

   o Allowing standard subscription parameter negotiation.

   o Allowing standard negotiation of SIP extensions.

   There are limitations on when it is appropriate to use the extension
   that allows an explicit subscription, related directly to definition
   of non-INVITE transaction handling SIP. These limitations are
   discussed in Section 4.1.

2. Conventions

   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].

3. Overview

   This section provides a non-normative overview of the behaviors
   defined in subsequent sections.

3.1. Explicit Subscriptions

   A SIP User Agent (UA) that wishes to issue a REFER request that will
   not create an implicit subscription, but will allow an explicit one,
   will include a new option tag, 'explicitsub', in the Require header
   field of the REFER request. This REFER could be sent either within
   an existing dialog or as an out-of-dialog request.

   If the recipient of the REFER accepts the request, it will begin
   managing the 'refer' event state described in RFC 3515 and will
   provide a URI that will reach an event server that will service
   subscriptions to that state. (In many cases, the recipient of the
   REFER will perform the role of event server itself.) That URI is
   returned in a new header field in the REFER response named 'Refer-
   Events-At'.

   The UA that issued the REFER can now subscribe to the 'refer' event
   at the provided URI, using a SUBSCRIBE request with a new dialog
   identifier. The full range of negotiation mechanisms is available
   for its use in that request. As detailed in RFCs 6665 and 3515, the
   event server accepting the subscription will send an immediate NOTIFY
   with the current refer event state, additional NOTIFY messages as the
   refer state changes, and a terminal NOTIFY message when the referred
   action is complete. It is, of course, possible that the initial
   NOTIFY is also the terminal NOTIFY.

   It is possible that the referred action is completed before the
   SUBSCRIBE arrives at the event server. The server needs to retain
   the final refer event state for some period of time to include in the
   terminal NOTIFY that will be sent for such subscriptions. It is also
   possible that a SUBSCRIBE will never arrive.

   This extension makes it possible to separate the event server that
   will handle subscriptions from the UA that accepted the REFER. Such
   a UA could use mechanisms such as PUBLISH [RFC3903] to convey the
   refer event state to the event server. This extension also makes it
   possible to allow more than one subscription to the refer event
   state.

3.2. No Subscriptions

   A UA that wishes to issue a REFER request that will not create an
   implicit subscription and wishes to tell the recipient that it is not
   interested in creating an explicit subscription will include a new
   option tag, 'nosub', in the Require header field of the REFER
   request. This REFER could be sent either within an existing dialog
   or as an out-of-dialog request.

   If the recipient of the REFER accepts the request, it knows not to
   create an implicit subscription and knows that no explicit
   subscription will be forthcoming. The recipient will continue to
   process the request indicated in the Refer-To header field as
   specified in RFC 3515, but it can avoid the cost of preparing to
   handle any 'refer' event subscriptions related to this REFER request.

4. The Explicit Subscription Extension

4.1. Sending a REFER

   To suppress the creation of any implicit subscription, and allow for
   an explicit one, a UA forming a REFER request will include the option
   tag 'explicitsub' in the Require header field of the request. The
   REFER request is otherwise formed following the requirements of
   [RFC3515]. Since this REFER has no chance of creating an implicit
   subscription, the UA MAY send the REFER request within an existing
   dialog or out-of-dialog.

   Note that if the REFER forks (see [RFC3261]), only one final response
   will be returned to the issuing UA. If it is important that the UA
   be able to subscribe to any refer state generated by accepting this
   request, the UA needs to form the request so that it will only be
   accepted in one place. This can be achieved by sending the REFER
   request within an existing dialog or by using the 'Target-Dialog'
   mechanism defined in [RFC4538]. If it is possible for the request to
   be accepted in more than one location, and things would go wrong if
   the UA did not learn about each location that the request was
   accepted, using this extension is not appropriate.

4.2. Processing a REFER Response

   The UA will process responses to the REFER request as specified in
   [RFC3515] (and, consequently, [RFC3261]). In particular, if the
   REFER was sent to an element that does not support or is unwilling to
   use this extension, the response will contain a 420 (Bad Extension)
   response code (see Section 8.1.3.5 of [RFC3261]). As that document
   states, the UA can retry the request without using this extension.

If the UA receives a 2xx-class response, it will contain a Refer-
Events-At header field (Section 4.8) with a single URI as its value.
If the UA is interested in the state of the referenced action, it
will subscribe to the 'refer' event at that URI.

4.3. Processing a Received REFER

An element receiving a REFER request requiring the 'explicitsub'
extension will use the same admissions policies that are used without
the extension, with the addition that it is acceptable to admit an
in-dialog REFER request requiring this extension since it cannot
create another usage inside that dialog. In particular, see
Section 5.2 of [RFC3515].

Accepting a REFER request that requires 'explicitsub' does not create
a dialog or a new usage within an existing dialog. The element MUST
NOT create an implicit subscription when accepting the REFER request.

If the REFER request was received within an existing dialog, the
accepting element will not be acting as a SIP-Events notifier in the
context of that dialog. If it is not otherwise subject to becoming a
notifier in the context of the dialog, none of the requirements in
[RFC6665], particularly the requirement to provide a Globally
Routable User Agent URI (GRUU) as the local contact, apply to the
message accepting the REFER request.

An element that accepts a REFER request with 'explicitsub' in its
Require header field MUST return a 200 response containing a sip: or
sips: URI in the Refer-Events-At header field that can be used to
subscribe to the refer event state associated with this REFER
request. This URI MUST uniquely identify this refer event state.
The URI needs to reach the event server when used in a SUBSCRIBE
request from the element that sent the REFER. One good way to ensure
the URI provided has that property is to use a GRUU [RFC5627] for the
event server. As discussed in Section 8, possession of this URI is
often the only requirement for authorizing a subscription to it.
Implementations SHOULD provide a URI constructed in a way that is
hard to guess. Again, using a GRUU (specifically, a temporary GRUU)
is one good way to achieve this property.

The accepting element will otherwise proceed with the processing
defined in [RFC3515].

The event server identified by the Refer-Events-At URI could receive
SUBSCRIBE requests at any point after the response containing the
Refer-Events-At header field is sent. Implementations should take
care to ensure the event server is ready to receive those SUBSCRIBE
requests before sending the REFER response, but as with all non-

INVITE responses, the response should be sent as soon as possible
(see [RFC4321]). It is also possible that the referred action may
complete before any SUBSCRIBE request arrives. The event server will
need to maintain the final refer event state for a period of time
after the action completes in order to serve such subscriptions (see
Section 4.7).

4.4. Subscribing to the 'refer' Event

A UA that possesses a URI obtained from a Refer-Events-At header
field MAY subscribe to the refer event state at that URI. It does so
following the requirements of [RFC6665], placing the token 'refer' in
the Event header field and the URI in the Request-URI of the
SUBSCRIBE request. The SUBSCRIBE request MUST NOT reuse any existing
dialog identifiers.

Subsequent handling of the subscription MUST follow the requirements
of [RFC6665] and [RFC3515]. In particular, as discussed in
Section 2.4.6 of [RFC3515], the NOTIFY messages in the subscription
might include an id parameter in their Event header fields.
Subsequent SUBSCRIBE requests used to refresh or terminate this
subscription MUST contain this id parameter. Note that the rationale
for the id parameter provided in that section is not relevant when
this extension is used. The URI returned in the Refer-Events-At
header field uniquely identifies appropriate state, making the id
parameter redundant. However, this behavioral requirement is
preserved to reduce the number of changes to existing implementations
in order to support this extension and to make it more likely that
existing diagnostic tools will work with little or no modification.

4.5. Processing a Received SUBSCRIBE

An event server receiving a SUBSCRIBE request will process it
according to the requirements of [RFC6665]. The event server MAY
choose to authorize the SUBSCRIBE request based on the Request-URI
corresponding to existing refer event state. It MAY also require
further authorization as discussed in Section 8.

When accepting a subscription, the event server will establish the
initial subscription duration using the guidance in Section 3.4 of
[RFC3515].

4.6. Sending a NOTIFY

NOTIFY messages within a subscription are formed and sent following
the requirements in [RFC3515]. See, in particular, Section 2.4.5 of
that document.

4.7. Managing 'refer' Event State

   As described in [RFC3515], an element creates the state for event
   'refer' when it accepts a REFER request. It updates that state as
   the referred request proceeds, ultimately reaching a state where the
   request has completed and the final state is known.

   In RFC 3515 implementations, it was a reasonable design choice to
   destroy the refer event state immediately after sending the NOTIFY
   that terminated the implicit subscription. This is not the case when
   using this extension. It is possible for the referenced request to
   complete very quickly, perhaps sooner than the time it takes the
   response to the REFER to traverse the network to the UA that sent the
   request and the time it takes that agent to send the SUBSCRIBE
   request for the event state to the URI the response provides. Thus,
   the event server MUST retain the final refer event state for a
   reasonable period of time, which SHOULD be at least 2*64*T1 (that is,
   64 seconds), representing an upper-bound estimate of the time it
   would take to complete two non-INVITE transactions: the REFER and an
   immediate SUBSCRIBE.

   If an otherwise acceptable SUBSCRIBE arrives during this retention
   period, the subscription would be accepted and immediately terminated
   with a NOTIFY containing the final event state with a Subscription-
   State of terminated with a reason value of "noresource".

4.8. The Refer-Events-At Header Field

   The Refer-Events-At header field is an extension-header as defined by
   [RFC3261]. Its ABNF [RFC5234] is as follows:

   Refer-Events-At = "Refer-Events-At" HCOLON
                        LAQUOT (SIP-URI / SIPS-URI) RAQUOT
                        *(SEMI generic-param)

   See [RFC3261] for the definition of the elements used in that
   production.

   Note that this rule does not allow a full addr-spec as defined in RFC
   3261, and it mandates the use of the angle brackets. That is:

   Refer-Events-At: <sips:vPT3izGmo8NTxaPADRZvEAY22BKx@example.com;gr>

   is well formed, but

   Refer-Events-At: sip:wsXa9mkHtPcGu8@example.com

   is invalid.

The Refer-Events-At header field is only meaningful in a 2xx-class
response to a REFER request. If it appears in the header of any
other SIP message, its meaning is undefined, and it MUST be ignored.

5. The No Subscription Extension

5.1. Sending a REFER

To suppress the creation of any implicit subscription and signal that
no explicit subscription will be forthcoming, a UA forming a REFER
request will include the option tag 'nosub' in the Require header
field of the request. The REFER request is otherwise formed
following the requirements of [RFC3515]. Since this REFER has no
chance of creating an implicit subscription, the UA MAY send the
REFER request within an existing dialog or out-of-dialog.

5.2. Processing a REFER Response

The UA will process responses to the REFER request as specified in
[RFC3515] (and, consequently, [RFC3261]). In particular, if the
REFER was sent to an element that does not support or is unwilling to
use this extension, the response will contain a 420 (Bad Extension)
response code (see Section 8.1.3.5 of [RFC3261]). As that document
states, the UA can retry the request without using this extension.

5.3. Processing a Received REFER

An element receiving a REFER request requiring the 'nosub' extension
will use the same admissions policies that would be used without the
extension, with the addition that it is acceptable to admit an in-
dialog REFER request requiring this extension since it cannot create
another usage inside that dialog. In particular, see Section 5.2 of
[RFC3515].

Accepting a REFER request that requires 'nosub' does not create a
dialog or a new usage within an existing dialog. The element MUST
NOT create an implicit subscription when accepting the REFER request.
Furthermore, the element accepting the REFER request is not required
to maintain any state for serving refer event subscriptions.

If the REFER is received within an existing dialog, the accepting
element will not be acting as a SIP-Events notifier in the context of
that dialog. If it is not otherwise subject to becoming a notifier
in the context of the dialog, none of the requirements in [RFC6665],
particularly the requirement to provide a GRUU as the local contact,
apply to the message accepting the REFER request.

The accepting element will otherwise proceed with the processing
defined in [RFC3515].

6. The 'explicitsub' and 'nosub' Option Tags

This document defines the 'explicitsub' option tag, used to signal
the use of the extension defined in Section 4, and the 'nosub' option
tag, used to signal the use of the extension defined in Section 5.

The use of either option tag in a Require header field is only
defined when it appears in a REFER request or a response to a REFER
request. A UA MUST NOT include the 'explicitsub' or 'nosub' option
tag in the Require header field of any request other than REFER. A
UA MUST NOT include the 'explicitsub' or 'nosub' option tag in the
Require header field of any SIP response other than a 200 or 421
response to a REFER request.

The 'explicitsub' and 'nosub' option tags MAY appear in the Supported
header field of SIP messages and in the sip.extensions feature tag
defined in [RFC3840]. This signals only that the UA including the
value is aware of the extensions. In particular, a UA can only
invoke the use of one of the extensions in a request. A UA MUST NOT
include either option tag in the Require header field of a 200
response to a REFER request if that tag was not present in the
Require header field of the request. A User Agent Server (UAS) that
is processing a REFER request that lists 'explicitsub' or 'nosub' in
its Supported header field and wishes to use one of those extensions
will return a 421 response indicating which extension is required.

7. Updates to RFC 3515

The requirement in Section 2.4.4 of [RFC3515] to reject out-of-dialog
SUBSCRIBE requests to event 'refer' is removed. An element MAY
accept a SUBSCRIBE request to event 'refer', following the
requirements and guidance in this document. REFER is no longer the
only mechanism that can create a subscription to event 'refer'.

8. Security Considerations

The security considerations of [RFC3515] all still apply to a REFER
request using this extension. The security considerations there for
the implicit subscription apply to any explicit subscription for the
'refer' event.

This update to RFC 3515 introduces a new authorization consideration.
An element receiving an initial SUBSCRIBE request to the 'refer'
event needs to decide whether the subscriber should be allowed to see
the refer event state. In RFC 3515, this decision was conflated with

accepting the REFER request, and the only possible subscriber was the
element that sent the REFER. With this update, there may be multiple
subscribers to any given refer event state.

This document allows an element to accept an initial SUBSCRIBE
request based on having a Request-URI that identifies existing refer
event state. (Such a URI will have previously been sent in the
Refer-Events-At header field in a successful REFER response). The
element retrieving that URI from the response and any elements that
element shares the URI with are authorized to SUBSCRIBE to the event
state. Consequently, the URI should be constructed so that it is not
easy to guess and should be protected against eavesdroppers when
transmitted. [RFC3261] details mechanisms for providing such
protection, such as sending SIP messages over Transport Layer
Security (TLS) or Datagram TLS (DTLS). See the Security
Considerations section of [RFC3261] for considerations when using
other security mechanisms. An event server receiving a REFER request
over an unprotected transport can redirect the requester to use a
protected transport before accepting the request. A good way to
ensure that subscriptions use a protected transport is to only
construct sips: URIs. The event server can also require any of the
additional authorization mechanisms allowed for any SIP request. For
example, the event server could require a valid assertion of the
subscriber's identity using [RFC4474].

The URI provided in a Refer-Events-At header field will be used as
the Request-URI of SUBSCRIBE requests. A malicious agent could take
advantage of being able to choose this URI in ways similar to the
ways an agent sending a REFER request can take advantage of the
Refer-To URI, as described in the Security Considerations section of
[RFC3515]. In particular, the malicious agent could cause a SIP
SUBSCRIBE to be sent as raw traffic towards a victim. If the victim
is not SIP aware and the SUBSCRIBE is sent over UDP, there is (at
most) a factor of 11 amplification due to retransmissions of the
request. The potential for abuse in this situation is lower than
that of the Refer-To URI, since the URI can only have a sip: or sips:
scheme, and is only provided in a REFER response. A malicious agent
would have to first receive a REFER request to take advantage of
providing a Refer-Events-At URI.

9. IANA Considerations

9.1. Register the 'explicitsub' Option Tag

   The option tag 'explicitsub' has been registered in the 'Option Tags'
   subregistry of the 'Session Initiation Protocol (SIP) Parameters'
   registry by adding a row with these values:

   Name: explicitsub

   Description: This option tag identifies an extension to REFER to
   suppress the implicit subscription and provide a URI for an explicit
   subscription.

   Reference: RFC 7614 (this document)

9.2. Register the 'nosub' Option Tag

   The option tag 'nosub' has been registered in the 'Option Tags'
   subregistry of the 'Session Initiation Protocol (SIP) Parameters'
   registry by adding a row with these values:

   Name: nosub

   Description: This option tag identifies an extension to REFER to
   suppress the implicit subscription and indicate that no explicit
   subscription is forthcoming.

   Reference: RFC 7614 (this document)

9.3. Register the Refer-Events-At Header Field

   The header field described in Section 4.8 has been registered in the
   'Header Fields' subregistry of the 'Session Initiation Protocol (SIP)
   Parameters' registry by adding a row with these values:

   Header Name: Refer-Events-At

   compact: none

   Reference: RFC 7614 (this document)

10. References

10.1. Normative References

   [RFC2119]   Bradner, S., "Key words for use in RFCs to Indicate
               Requirement Levels", BCP 14, RFC 2119,
               DOI 10.17487/RFC2119, March 1997,
               <http://www.rfc-editor.org/info/rfc2119>.

   [RFC3261]   Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston,
               A., Peterson, J., Sparks, R., Handley, M., and E.
               Schooler, "SIP: Session Initiation Protocol", RFC 3261,
               DOI 10.17487/RFC3261, June 2002,
               <http://www.rfc-editor.org/info/rfc3261>.

   [RFC3515]   Sparks, R., "The Session Initiation Protocol (SIP) Refer
               Method", RFC 3515, DOI 10.17487/RFC3515, April 2003,
               <http://www.rfc-editor.org/info/rfc3515>.

   [RFC3840]   Rosenberg, J., Schulzrinne, H., and P. Kyzivat,
               "Indicating User Agent Capabilities in the Session
               Initiation Protocol (SIP)", RFC 3840,
               DOI 10.17487/RFC3840, August 2004,
               <http://www.rfc-editor.org/info/rfc3840>.

   [RFC5234]   Crocker, D., Ed. and P. Overell, "Augmented BNF for Syntax
               Specifications: ABNF", STD 68, RFC 5234,
               DOI 10.17487/RFC5234, January 2008,
               <http://www.rfc-editor.org/info/rfc5234>.

   [RFC6665]   Roach, A., "SIP-Specific Event Notification", RFC 6665,
               DOI 10.17487/RFC6665, July 2012,
               <http://www.rfc-editor.org/info/rfc6665>.

10.2. Informative References

   [RFC3903]   Niemi, A., Ed., "Session Initiation Protocol (SIP)
               Extension for Event State Publication", RFC 3903,
               DOI 10.17487/RFC3903, October 2004,
               <http://www.rfc-editor.org/info/rfc3903>.

   [RFC4321]   Sparks, R., "Problems Identified Associated with the
               Session Initiation Protocol's (SIP) Non-INVITE
               Transaction", RFC 4321, DOI 10.17487/RFC4321, January
               2006, <http://www.rfc-editor.org/info/rfc4321>.

   [RFC4474]   Peterson, J. and C. Jennings, "Enhancements for
               Authenticated Identity Management in the Session
               Initiation Protocol (SIP)", RFC 4474,
               DOI 10.17487/RFC4474, August 2006,
               <http://www.rfc-editor.org/info/rfc4474>.

   [RFC4538]   Rosenberg, J., "Request Authorization through Dialog
               Identification in the Session Initiation Protocol (SIP)",
               RFC 4538, DOI 10.17487/RFC4538, June 2006,
               <http://www.rfc-editor.org/info/rfc4538>.

   [RFC5057]   Sparks, R., "Multiple Dialog Usages in the Session
               Initiation Protocol", RFC 5057, DOI 10.17487/RFC5057,
               November 2007, <http://www.rfc-editor.org/info/rfc5057>.

   [RFC5589]   Sparks, R., Johnston, A., Ed., and D. Petrie, "Session
               Initiation Protocol (SIP) Call Control - Transfer",
               BCP 149, RFC 5589, DOI 10.17487/RFC5589, June 2009,
               <http://www.rfc-editor.org/info/rfc5589>.

   [RFC5627]   Rosenberg, J., "Obtaining and Using Globally Routable User
               Agent URIs (GRUUs) in the Session Initiation Protocol
               (SIP)", RFC 5627, DOI 10.17487/RFC5627, October 2009,
               <http://www.rfc-editor.org/info/rfc5627>.

Author's Address

   Robert Sparks
   Oracle
   7460 Warren Parkway, Suite 300
   Frisco, Texas 75034
   United States

   Email: rjsparks@nostrum.com