SIPPING Working Group
Internet-Draft
Expires: November 9, 2004

M. Garcia-Martin
Nokia
G. Camarillo
Ericsson
May 11, 2004

Multiple recipient MESSAGE requests in the Session Initiation Protocol (SIP)
draft-garcia-sipping-message-exploder-00.txt

Abstract

This document specifies how to request a MESSAGE exploder to send a copy of a MESSAGE to a set of destinations. The client sends a SIP MESSAGE request with a URI list to the MESSAGE exploder, which sends a similar MESSAGE request to each of URIs included in the list.

Table of Contents

1. Introduction

SIP [2] can carry instant messages in MESSAGE [3] requests. The Advanced Instant Messaging Requirements for SIP [6] mentions the need for sending a MESSAGE request to multiple receipients:

"REQ-GROUP-3: It MUST be possible for a user to send to an ad-hoc group, where the identities of the recipients are carried in the message itself."

To meet this requirement, we allow SIP MESSAGE requests carry an URI list as specified in [4]. The Request-URI of the MESSAGE request contains a "list" URI parameter that points to a body part that carries the URI list. A specialized application server receives the request and sends a similar MESSAGE request to each of the URIs in the list. Each of these MESSAGE requests contains a copy of the body included in the original MESSAGE request.

The UAC needs to be configured with the SIP URI of the application server that provides the functionality. Discovering and provisioning of this URI to the UAC is outside the scope of this document.

2. Terminology

In this document, the key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" are to be interpreted as described in BCP 14, RFC 2119 [1] and indicate requirement levels for compliant implementations.

'MESSAGE exploder': SIP application server that receives a MESSAGE request with a URI list and sends a similar MESSAGE request to each URI in the list. MESSAGE exploders behave effectively as specialised B2BUAs (Back-To-Back-User-Agents). A MESSAGE exploder can be modelled as a fractional function of a B2BUA that can offer other exploder functionality (e.g., for other SIP methods), although that other exploder functionality is outside the scope of this document. In this document we only discuss the explosion of SIP MESSAGE requests.

'Incoming MESSAGE request': A SIP MESSAGE request that a UAC creates and addresses to a SIP MESSAGE exploder. Besides the regular instant message payload, an incoming MESSAGE request contains a URI list.

'Outgoing MESSAGE request': A SIP MESSAGE request that a MESSAGE exploder creates and addresses to a UAS. It contains the regular instant message payload.

3. Procedures at the UAC

A client that wants to create a multiple recipient MESSAGE request SHOULD add a "list" parameter (specified in [4]) to the MESSAGE exploder's URI and MUST place the resulting URI in the Request-URI of the MESSAGE request. The "list" parameter MUST contain a pointer to a URI list that contains the recipients of the MESSAGE. The following is an example of a Request-URI with a "list" parameter.

sip:message-exploder.example.com;list=cid:cn35t8jf@uac.example.com

Multiple recipient MESSAGE requests will typically contain a multiparty body that contains the body carrying the list and the actual instant message payload. In some cases, the MESSAGE request will contain bodies other than the text and the list bodies, for instance, when the request is protected with S/MIME.

Typically the MESSAGE exploder will copy all the significant header fields in the exploded MESSAGE request. However, there might be cases where the SIP UA wants the MESSAGE exploder to add a particular header field with a particular value, when the header field wasn't present in the MESSAGE request sent by the UAC. In this case the UAC MAY use the "?" mechanism described in Section 19.1.1 of RFC 3261 [2] to encode extra information in any URI in the list. However, the UAC MUST NOT use the special "body" hname (see Section 19.1.1 of RFC 3261 [2]) to encode a body, since the body is present in the MESSAGE request itself.

The following is an example of a URI that uses the "?" mechanism:

sip:message-exploder.example.com;list=cid:cn35t8jf@uac.example.com?
Accept-Contact=*%3bmobility%3d%22mobile%22

The previous URI requests the MESSAGE exploder to add the following header field to a MESSAGE request:

Accept-Contact: *;mobility="mobile"

As described in [4], the default format for URI lists in SIP is the XCAP resource list format [5]. User Agents compliant to this specification MUST support the XCAP resource list format [5] and MAY support other formats.

UAs generating multiple recipient MESSAGEs SHOULD use flat lists (i.e., no hierarchical lists), SHOULD NOT use any entry's attributes but "uri", and SHOULD NOT include any elements inside entries but "display-name" elements.

4. Procedures at the MESSAGE exploder

On receiving a MESSAGE request that contains a "list" parameter in the Request-URI as described in [4], a MESSAGE exploder SHOULD answer to the UAC with a 202 Accepted response. Note that the status code in the response to the MESSAGE does not provide any information about whether or not the MESSAGEs generated by the exploder were successfully delivered to the URIs in the list. That is, a 202 Accepted means that the MESSAGE exploder has received the MESSAGE and that it will try to send a similar MESSAGE to the URIs in the list. Designing a mechanism to inform a client about the delivery status of an instant message is outside the scope of this document.

On receiving a MESSAGE request that contains a "list" parameter in the Request-URI as described [4], a MESSAGE exploder SHOULD create as many new MESSAGE requests as URIs the list contains, except when two of those URIs are equivalent (section 19.1.4 of RFC 3261 [2] defines equivalent URIs), in which case the MESSAGE exploder SHOULD create only one outgoing MESSAGE request per URI.

The Request-URI of each of the outgoing MESSAGE requests MUST NOT include any "list" parameters. This avoids loops in exploding MESSAGE requests. It also avoids the implementation a MESSAGE exploder functionality in the UAS, that otherwise, would be required.

On creating the body of each of the outgoing MESSAGE requests, the MESSAGE exploder tries to keep the relevant bodies of the incoming MESSAGE request and copies them to the outgoing MESSAGE request. The following guidelines are provided:

- The incoming MESSAGE request typically contains a URI list body [4] with the actual list of recipients. The MESSAGE exploder need not copy the URI list body to the outgoing MESSAGE request, although it MAY do it.
  NOTE: This document does not provide any semantics associated to a URI list body included in an outgoing MESSAGE request. Future extensions can indicate actions at a UAS when it receives that body.
- The MESSAGE exploder MUST NOT copy any security body (such as an S/MIME signed body) addressed to the MESSAGE exploder to the outgoing MESSAGE request. This includes, e.g., security bodies signed with the public key of the exploder.
- The MESSAGE exploder SHOULD copy all the rest of the message bodies (e.g., text messages, images, etc.) to the outgoing MESSAGE request.
- If there is only one body left, the MESSAGE exploder MUST remove the multipart/mixed wrapper in the outgoing MESSAGE request.

The rest of the MESSAGE request corresponding to a given URI in the list MUST be created following the rules in Section 19.1.5 "Forming Requests from a URI" of RFC 3261 [2]. In particular, Section 19.1.5 of RFC 3261 [2] states:

"An implementation SHOULD treat the presence of any headers or body parts in the URI as a desire to include them in the message, and choose to honor the request on a per-component basis."

SIP allows to append a "method" parameter to a URI. Therefore, it is legitimate that an the "uri" attribute of the "entry" element in the XCAP resource list contains a "method" parameter. MESSAGE exploders MUST generate only MESSAGE requests, regardless of the "method" parameter that the URIs in the list indicate. Effectively, MESSAGE exploders MUST ignore the "method" parameter in each of the URIs present in the URI list.

It is RECOMMENDED that the MESSAGE exploder copies the value From header field of the incoming MESSAGE into the outgoing MESSAGE requests (note that this need not apply to the "tag" parameter). The MESSAGE exploder SHOULD also copy to the outgoing MESSAGE request any P-Asserted-Identity header fields that can be present in the incoming MESSAGE request.

On each given outgoing MESSAGE request, the MESSAGE exploder SHOULD generate a new To header field value which, according to the procedures of RFC 3261 Section 8.1.1.1, should be equal to the Request-URI of the outgoing MESSAGE request.

On each given outgoing MESSAGE request, the MESSAGE exploder SHOULD initialize the values of the Call-ID, CSeq and Max-Forwards header fields. The MESSAGE exploder should also include its own value in the Via header field.

A MESSAGE exploder receiving a URI list with more information than what we have just described SHOULD discard all the extra information.

As described in [4], the default format for URI lists in SIP is the XCAP resource list format [5]. MESSAGE exploders compliant to this specification MUST support the XCAP resource list format [5] and MAY support other formats.

5. Examples

The following is an example of an incoming MESSAGE request which carries a URI list in its body.

MESSAGE sip:exploder.example.com;list=cid:cn35t8jf@uac.example.com

```
        SIP/2.0
   Via: SIP/2.0/TCP uac.example.com
       ;branch=z9hG4bKhjhs8ass83
   Max-Forwards: 70
   To: MESSAGE Exploder <sip:exploder.example.com>
   From: Carol <sip:carol@example.com>;tag=32331
   Call-ID: d432fa84b4c76e66710
   CSeq: 1 MESSAGE
   Content-Type: multipart/mixed;boundary="boundary1"
   Content-Length: xxx

   --boundary1
   Content-Type: text/plain
   Content-Length: 13

   Hello World!

   --boundary1
   Content-Type: application/resource-lists+xml
   Content-Length: 315
   Content-ID: <cn35t8jf@uac.example.com>

   <?xml version="1.0" encoding="UTF-8"?>
   <resource-lists xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
     <list>
       <entry uri="sip:bill@example.com" />
       <entry uri="sip:joe@example.com" />
       <entry uri="sip:ted@example.com" />
       <entry uri="sip:bob@example.com" />
     </list>
   </resource-lists>
   --boundary1--
```

Figure 4: Multiple recipient incoming MESSAGE request

The following is an example of one of the outgoing MESSAGE requests that the MESSAGE exploder creates.

```
MESSAGE sip:bill@example.com SIP/2.0
Via: SIP/2.0/TCP exploder.example.com
    ;branch=z9hG4bKhjhs8as34sc
Max-Forwards: 70
To: <sip:bill@example.com>
From: Carol <sip:carol@uac.example.com>;tag=210342
Call-ID: 39s02sdsl20d9sj2l
CSeq: 1 MESSAGE
Content-Type: text/plain
Content-Length: 13

Hello World!
```

Figure 5: Outgoing MESSAGE request

6. Security Considerations

If MESSAGE exploders are not implemented properly, they could become a SPAM amplification tool. The SPAMMER would have the exploder, which will generally have a higher access bandwidth and more processing power, send a SPAM message to a large set of destinations. This section provides guidelines to prevent SPAM amplifications in particular, and DoS attacks in general. In addition, we describe how to provide content confidentiality and integrity.

MESSAGE exploders MUST authenticate and authorize any user agent sending a multiple recipient MESSAGE. Additionally, MESSAGE exploders MAY have policies that limit the number of URIs in the list, as a very long list could be used in a DoS attack to place a large burden on the exploder to send a large number of MESSAGEs or to perform an amplification attack.

In case an exploder is used to send unsolicited instant messages (i.e., SPAM), it should be possible to track down the sender of such messages. To do that, MESSAGE exploders MAY provide information about the identity of the original sender of the MESSAGE in their outgoing MESSAGE requests. Exploders can use Authenticated Identity Bodies (AIB) [7] or P-Asserted-Identity header fields [8] to provide this information. Furthermore, it is RECOMMENDED that MESSAGE exploders keep a log of all the transactions they handle (for a reasonable period of time), so that SPAMMERS can be tracked down.

It is RECOMMENDED that user agents using MESSAGE exploders integrity protect the contents of their instant messages and the list of recipients using S/MIME. If the contents of the instant message or the list of recipients needs to be kept private, the user agent SHOULD also use S/MIME to prevent a third party from viewing this

information.

7. Acknowledgements

Duncan Mills supported the idea of having 1 to n MESSAGEs. Ben Campbell and Paul Kyzivat provided helpful comments.

8. Change control

8.1 Changes from draft-garcia-simple-message-exploder-00.txt to draft-garcia-sipping-message-exploder-00.txt

The MESSAGE exploder may or may not copy the URI list body to the outgoing MESSAGE request. This allows to extend the mechanism with a Reply-to-all feature.

It is clarified that the MESSAGE exploder must not include a list in the outgoing MESSAGE requests. This avoids loops or requires a MESSAGE exploder functionality in the next hop.

The MESSAGE exploder must remove the multipart/mixed wrapper if there is only one body left in the outgoing MESSAGE request.

Filename changed due to focus on the SIPPING WG.

9. References

9.1 Normative References

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

[2] 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.

[3] Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C. and D. Gurle, "Session Initiation Protocol (SIP) Extension for Instant Messaging", RFC 3428, December 2002.

[4] Camarillo, G., "Providing a Session Initiation Protocol (SIP) Application Server with a List of URIs", draft-camarillo-sipping-uri-list-00 (work in progress), November 2003.

[5] Rosenberg, J., "An Extensible Markup Language (XML) Configuration Access Protocol (XCAP) Usage for Presence Lists", draft-ietf-simple-xcap-list-usage-01 (work in progress), October

2003.

9.2 Informational References

[6] Rosenberg, J., "Advanced Instant Messaging Requirements for the Session Initiation Protocol (SIP)", draft-rosenberg-simple-messaging-requirements-00 (work in progress), December 2002.

[7] Peterson, J., "SIP Authenticated Identity Body (AIB) Format", draft-ietf-sip-authid-body-02 (work in progress), July 2003.

[8] Jennings, C., Peterson, J. and M. Watson, "Private Extensions to the Session Initiation Protocol (SIP) for Asserted Identity within Trusted Networks", RFC 3325, November 2002.

Authors' Addresses

Miguel A. Garcia-Martin
Nokia
P.O.Box 407
NOKIA GROUP, FIN 00045
Finland

EMail: miguel.an.garcia@nokia.com

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.