SIPPING Working Group
Internet-Draft
Expires: August 3, 2006

G. Camarillo
Ericsson
A. Roach
Estacado Systems
January 30, 2006

Framework and Security Considerations for Session Initiation Protocol (SIP) Uniform Resource Identifier (URI)-List Services
draft-ietf-sipping-uri-services-05.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the need for SIP URI-list services and provides requirements for their invocation. Additionaly, it defines a framework for SIP URI-List services which includes security considerations applicable to these services.

Table of Contents

1. Introduction

Some applications require that, at a given moment, a SIP [3] UA (User Agent) performs a similar transaction with a number of remote UAs. For example, an instant messaging application that needs to send a particular message (e.g., "Hello folks") to n receivers needs to send n MESSAGE requests; one to each receiver.

When the transacton that needs to be repeated consists of a large request, or the number of recipients is high, or both, the access network of the UA needs to carry a considerable amount of traffic. Completing all the transactions on a low-bandwidth access would require a long time. This is unacceptable for a number of applications.

A solution to this problem consists of introducing URI-list services in the network. The task of a SIP URI-list service is to receive a request that contains or references a URI-list (i.e., a list of one or more URIs) and send a number of similar requests to the destinations in this list. Once the requests are sent, the URI-list service typically informs the UA about their status. Effectively, the URI-list service behaves as a B2BUA (Back-To-Back-User-Agent).

A given URI-list service can take as an input a URI-list contained in the SIP request sent by the client or an external URI-list (e.g., the Request-URI is a SIP URI which is associated with a URI-list at the server). External URI-lists are typically set up using out-of-band mechanisms (e.g., XCAP [9]). An example of a URI-list service for SUBSCRIBE requests that uses stored URI-lists is described in [5].

The Advanced Instant Messaging Requirements for SIP [6] mentions the need for request-contained URI-list services for MESSAGE transactions:

"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."

The remainder of this document provides requirements and a framework for URI-list services using request-contained URI-lists, external URI-lists, or both.

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.

3. Requirements

Section 3.1 discusses requirements that only apply to URI-list services that use request-contained lists and Section 3.2 discusses requirements that also apply services using external lists.

3.1. Requirements for URI-List Services Using Request-Contained Lists

REQ 1: The URI-list service invocation mechanism MUST allow the invoker to provide a list of destination URIs to the URI-list service.
REQ 2: The invocation mechanism SHOULD NOT require more than one RTT (Round-Trip Time).

3.2. General Requirements for URI-List Services

GEN 1: A URI-list service MAY include services beyond sending requests to the URIs in the URI-list. That is, URI-list services can be modelled as application servers. For example, a URI-list service handling INVITE requests may behave as a conference server and perform media mixing for all the participants.
GEN 2: The interpretation of the meaning of the URI-list sent by the invoker MUST be at the discretion of the application to which the list is sent.
GEN 3: It MUST be possible for the invoker to find out about the result of the operations performed by the URI-list service with the URI-list. An invoker may, for instance, be interested in the status of the transactions initiated by the URI-list service.
GEN 4: URI-list services MUST NOT send requests to any destination without authenticating the invoker.

4. Framework

This framework is not restricted to application servers that only provide request fan-out services. Per GEN 1, this framework also deals with application servers that provide a particular service that includes a request fan-out (e.g., a conference server that INVITEs several participants which are chosen by a user agent).

4.1. Carrying URI-Lists in SIP

The requirements that relate to URI-list services that use request-contained lists identify the need for a mechanism to provide a SIP

URI-list service with a URI-list in a single RTT. We define a new disposition type [2] for the Content-Disposition header field: recipient-list. Both requests and responses MAY carry recipient-list bodies. Bodies whose disposition type is recipient-list carry a list of URIs that contains the final recipients of the requests to be generated by a URI-list service.

The default format for recipient-list bodies is service specific. So, URI-list services specifications MUST specify a default format for recipient-list bodies used within a particular service. In any case, clients SHOULD NOT include any particular URI more than once in a given URI-list.

A UA server receiving a request with more than one recipient-list body parts (e.g., each body part using a different URI-list format) MUST behave as if it had received a single URI-list which contains all the URIs present in the different body parts.

A UA server receiving a recipient-list URI-list which contains a URI more than once MUST behave as if that URI appeared in the URI-list only once. The UA server uses the comparison rules specific to the URI scheme of each of the URIs in the URI-list to determine if there is any URI which appears more than once.

The way a UA server receiving a URI-list interprets it is service specific, as described in Section 4.2.

4.2. Processing of URI-Lists

According to GEN 1 and GEN 2, URI-list services can behave as application servers. That is, taking a URI-list as an input, they can provide arbitrary services. So, the interpretation of the URI-list by the server depends on the service to be provided. For example, for a conference server, the URIs in the list may identify the initial set of participants. On the other hand, for a server dealing with MESSAGEs, the URIs in the list may identify the recipients of an instant message.

At the SIP level, this implies that the behavior of application servers receiving requests with URI-lists SHOULD be specified on a per service basis. Examples of such specifications are [10] for INVITE, [11] for REFER, [12] for MESSAGE, and [13] for SUBSCRIBE.

4.3. Results

According to GEN 3, user agents should have a way to obtain information about the operations performed by the application server. Since these operations are service specific, the way user agents are

kept informed is also service specific. For example, a user agent establishing an adhoc conference with an INVITE with a URI-list may discover which participants were successfully brought in into the conference by using the conference package [8].

5. Security Considerations

Security plays an important role in the implementation of any URI-list service. In fact, it is the most important common area across all types of URI-list services.

By definition, a URI-list service takes one request in and sends a potentially large number of them out. Attackers may attempt to use URI-list services as traffic amplifiers to launch DoS (Denial of Service) attacks. This section provides guidelines to avoid these attacks.

5.1. List Integrity and Confidentiality

Attackers may attempt to modify URI-lists sent from clients to servers. This would cause a different behavior at the server than expected by the client (e.g., requests being sent to different recipients as the ones specified by the client). To prevent this attack, clients SHOULD integrity protect URI-lists using mechanisms such as S/MIME, which can also provide URI-list confidentiality if needed.

5.2. Amplification Attacks

URI-list services take a request in and send a potentially large number of them out. Given that URI-list services are typically implemented on top of powerful servers with high-bandwidth access links, we should be careful to keep attackers from using them as amplification tools to launch DoS (Denial of Service) attacks.

Attackers may attempt to send a URI-list containing URIs whose host parts route to the victims of the DoS attack. These victims do not need to be SIP nodes; they can be non-SIP endpoints or even routers. If this attack is successful, the result is that an attacker can flood with traffic a set of nodes, or a single node, without needing to generate a high volume of traffic itself.

> Note, in any case, that this problem is not specific to SIP URI-list services; it also appears in scenarios which relate to multihoming where a server needs to contact a set of IP addresses provided by a client (e.g., an SCTP [4] endpoint using HEARTBEATs to check the status of the IP addresses provided by its peer at

association establishment).

There are several measures that need to be taken to prevent this type of attack. The first one is keeping unauthorized users from using URI-list services. So, URI-list services MUST NOT perform any request explosion for an unauthorized user. URI-list services MUST authenticate users and check whether they are authorized to request the service before performing any request fan-out.

Note that the risk of this attack also exists when a client uses stored URI-lists. Application servers MUST use authentication and authorization mechanisms with equivalent security properties when dealing with stored and request-contained URI-lists.

Even though the previous rule keeps unauthorized users from using URI-list services, authorized users may still launch attacks using a these services. To prevent these attacks, we introduce the concept of opt-in lists. That is, URI-list services should not allow a client to place a user (identified by his or her URI) in a URI-list unless the user has previously agreed to be placed in such a URI-list. So, URI-list services MUST NOT send a request to a destination which has not agreed to receive requests from the URI-list service beforehand. Users can agree to receive requests from a URI-list service in several ways, such as filling a web page, sending an email, signing a contract, or using the Framework for Consent-Based Communications in SIP [14] (whose requirements are discussed in [15]). Additionally, users MUST be able to further describe the requests they are willing to receive. For example, a user may only want to receive requests from a particular URI-list service on behalf of a particular user. Effectively, these rules make URI-lists used by URI-list services opt-in lists.

When a URI-list service receives a request with a URI-list from a client, the URI-list service checks whether all the destinations have agreed beforehand to receive requests from the service on behalf of this client. If the URI-list has permission to send requests to all of the targets in the request, it does so. If not, the behavior of the URI-list service is service specific. It may only send requests to the targets it has permissions for or it may not send any request at all.

The Framework for Consent-Based Communications in SIP [14] specifies a means for the URI-list service to inform the client that some permissions were missing and how to request them.

Note that the mechanism used to obtain permissions should not create opportunities to launch DoS amplification attacks. These attacks would be possible if, for instance, the URI-list service automatically contacted the full set of targets for which it did not have permissions in order to request permissions. The URI-list service would be receiving one SIP request and sending out a number of authorization request messages. The Framework for Consent-Based Communications in SIP [14] avoids this type of attack by having the client generate roughly the same amount of traffic towards the URI-list service as the service generates towards the destinations.

In order to have an interoperable way to meet the requirements related to opt-in lists described in this section, URI-list services MUST implement, and SHOULD use, The Framework for Consent-Based Communications in SIP [14].

5.3. General Issues

URI-list services MAY have policies that limit the number of URIs in the lists they accept, as a very long list could be used in a denial of service attack to place a large burden on the URI-list service to send a large number of SIP requests.

The general requirement GEN 4, which states that URI-list services need to authenticate their clients, and the previous rules apply to URI-list services in general. In addition, specifications dealing with individual methods MUST describe the security issues that relate to each particular method.

6. IANA Considerations

This document defines a new Content-Disposition header field disposition type (recipient-list) in Section 4.1. This value should be registered in the IANA registry for Mail Content Disposition Values and Parameters with the following description:

recipient-list	the body contains a list of URIs

7. Acknowledges

Duncan Mills and Miguel A. Garcia-Martin supported the idea of 1 to n MESSAGEs. Jon Peterson, Dean Willis, and Jonathan Rosenberg provided useful comments.

8. References

8.1. Normative References

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

[2] Troost, R., Dorner, S., and K. Moore, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 1997.

[3] 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.

8.2. Informational References

[4] Bradner, S., "A Proposal for an MOU-Based ICANN Protocol Support Organization", RFC 2690, September 1999.

[5] Roach, A., Rosenberg, J., and B. Campbell, "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", draft-ietf-simple-event-list-07 (work in progress), January 2005.

[6] Rosenberg, J., "Advanced Instant Messaging Requirements for the Session Initiation Protocol (SIP)", draft-rosenberg-simple-messaging-requirements-01 (work in progress), February 2004.

[7] Peterson, J. and C. Jennings, "Enhancements for Authenticated Identity Management in the Session Initiation Protocol (SIP)", draft-ietf-sip-identity-06 (work in progress), October 2005.

[8] Rosenberg, J., "A Session Initiation Protocol (SIP) Event Package for Conference State", draft-ietf-sipping-conference-package-12 (work in progress), July 2005.

[9] Rosenberg, J., "The Extensible Markup Language (XML) Configuration Access Protocol (XCAP)", draft-ietf-simple-xcap-08 (work in progress), October 2005.

[10] Camarillo, G. and A. Johnston, "Conference Establishment Using Request-Contained Lists in the Session Initiation Protocol (SIP)", draft-ietf-sipping-uri-list-conferencing-04 (work in progress), October 2005.

[11] Camarillo, G., "Refering to Multiple Resources in the Session Initiation Protocol (SIP)", draft-ietf-sipping-multiple-refer-04 (work in progress), October 2005.

[12] Garcia-Martin, M. and G. Camarillo, "Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)", draft-ietf-sipping-uri-list-message-05 (work in progress), January 2006.

[13] Camarillo, G., "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)", draft-ietf-sipping-uri-list-subscribe-04 (work in progress), October 2005.

[14] Rosenberg, J., "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)", draft-ietf-sipping-consent-framework-03 (work in progress), October 2005.

[15] Rosenberg, J., "Requirements for Consent-Based Communications in the Session Initiation Protocol (SIP)", draft-ietf-sipping-consent-reqs-04 (work in progress), January 2006.

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

Email: Gonzalo.Camarillo@ericsson.com

Adam Roach
Estacado Systems
Dallas, TX
US

Email: adam@estacado.net

Intellectual Property Statement

Disclaimer of Validity

Copyright Statement

Acknowledgment

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