Network Working Group
Request for Comments: 5363
Category: Standards Track

G. Camarillo
Ericsson
A.B. Roach
Tekelec

October 2008

Framework and Security Considerations for Session Initiation Protocol (SIP) URI-List Services

Status of This Memo

Abstract

This document describes the need for SIP URI-list services and provides requirements for their invocation. Additionally, 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 [RFC3261] 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 transaction that needs to be repeated consists of a large request, or when 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 that is associated with a URI list at the server). External URI lists are typically set up using out-of-band mechanisms (e.g., XML Configuration Access Protocol (XCAP) [RFC4825]). An example of a URI-list service for SUBSCRIBE requests that uses stored URI lists is described in [RFC4662].

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

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. 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 to 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 transaction.

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 modeled 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 that are chosen by a user agent).

4.1. Carrying URI Lists in SIP

The requirements related 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 transaction. We define a new disposition type [RFC2183] 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 that contains all the URIs present in the different body parts.

A UA server receiving a recipient-list URI list that 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 that appears more than once. Additionally, Section 4 of "Extensible Markup Language (XML) Format Extension for Representing Copy Control Attributes in Resource Lists" [RFC5364] discusses cases where duplicated URI entries are tagged with different values of the 'copyControl' attribute. Naturally, URI-list services using the 'copyControl' attribute defined in [RFC5364] need to follow the recommendations in [RFC5364] with respect to avoiding sending duplicated requests.

The way a UA server interprets a URI list that it has received 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 [RFC5366] for INVITE, [RFC5365] for MESSAGE, and [RFC5367] 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 ad hoc conference with an INVITE with a URI list may discover which participants were successfully brought into the conference by using the conference package [RFC4575].

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 than the ones specified by the client). To prevent this attack, clients SHOULD integrity protect URI lists using end-to-end mechanisms such as S/MIME or, if not available, hop-by-hop mechanisms such as TLS. Both S/MIME and TLS 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 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 a set of nodes, or a single node, with traffic without needing to generate a high volume of traffic itself.

In any case, note that this problem is not specific to SIP URI-list services; it also appears in scenarios that relate to multihoming where a server needs to contact a set of IP addresses provided by a client.

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 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 that 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 "A Framework for Consent-Based Communications in the Session Initiation Protocol (SIP)" [RFC5360], whose requirements are discussed in [RFC4453]. 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 that used by URI-list services into 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, it does not send any request at all.

The Framework for Consent-Based Communications in SIP [RFC5360] 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 [RFC5360] 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 "A Framework for Consent-Based Communications in SIP" [RFC5360].

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.

A URI-list service generates a set of requests from a URI list. Section 19.1.5 of [RFC3261] provides recommendations that need to be taken into consideration when forming a request from a URI. Naturally, those recommendations apply to all SIP URI-list services.

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 has been registered in the IANA registry for Mail Content Disposition Values and Parameters with the following description:

recipient-list	The body includes a list of URIs to which URI-list services are to be applied.

7. Acknowledgements

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

8. References

8.1. Normative References

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

[RFC2183] Troost, R., Dorner, S., and K. Moore, "Communicating Presentation Information in Internet Messages: The Content-Disposition Header Field", RFC 2183, August 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.

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

8.2. Informative References

[RFC4453] Rosenberg, J., Camarillo, G., and D. Willis, "Requirements for Consent-Based Communications in the Session Initiation Protocol (SIP)", RFC 4453, April 2006.

[RFC4575] Rosenberg, J., Schulzrinne, H., and O. Levin, "A Session Initiation Protocol (SIP) Event Package for Conference State", RFC 4575, August 2006.

[RFC4662] Roach, A.B., Campbell, B., and J. Rosenberg, "A Session Initiation Protocol (SIP) Event Notification Extension for Resource Lists", RFC 4662, August 2006.

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

[RFC5364] Garcia-Martin, M. and G. Camarillo, "Extensible Markup Language (XML) Format Extension for Representing Copy Control Attributes in Resource Lists", RFC 5364, October 2008.

[RFC5365] Garcia-Martin, M. and G. Camarillo, "Multiple-Recipient MESSAGE Requests in the Session Initiation Protocol (SIP)", RFC 5365, October 2008.

[RFC5366] Camarillo, G. and A. Johnston, "Conference Establishment Using Request-Contained Lists in the Session Initiation Protocol (SIP)", RFC 5366, October 2008.

[RFC5367] Camarillo, G., Roach, A.B., and O. Levin, "Subscriptions to Request-Contained Resource Lists in the Session Initiation Protocol (SIP)", RFC 5367, October 2008.

Authors' Addresses

Gonzalo Camarillo
Ericsson
Hirsalantie 11
Jorvas 02420
Finland

EMail: Gonzalo.Camarillo@ericsson.com

Adam Roach
Tekelec
17210 Campbell Rd Ste 250
Dallas, TX 75252
USA

EMail: Adam.Roach@tekelec.com

Full Copyright Statement

Intellectual Property