SIPPING   S. Olson
Internet-Draft   Microsoft
Expires: March 24, 2003   September 23, 2002

Requirements for Content Indirection in Session Initiation Protocol (SIP) Messages

draft-ietf-sipping-content-indirect-02

Status of this Memo

This Internet-Draft will expire on March 24, 2003.

Copyright Notice

Abstract

This specification defines requirements for a mechanism to indirectly specify the content of a SIP message for the purpose of transferring the content via a non-SIP channel.

1. 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 RFC 2119 [1].

2. Introduction

The purpose of the Session Initiation Protocol [2] (SIP) is to create, modify, or terminate sessions with one or more participants. SIP messages, like HTTP, are sytnactically composed of a start line, one or more headers, and an optional body. Unlike HTTP, SIP is not designed as a general purpose transport of data.

There are numerous reasons why it might be desirable to indirectly specify the content of a SIP message body. For bandwidth limited applications such as cellular wireless, indirection provides a means to annotate the (indirect) content with meta-data which may be used by the recipient to determine whether or not to retrieve the content over the resource limited link.

It is also possible that the content size to be transferred might potentially overwhelm intermediate signaling proxies, thereby unnecessarily increasing network latency. For time-sensitive SIP applications, this may be unacceptable. Indirect content can remedy this by moving the transfer of this content out of the SIP signaling network and into a potentially separate data transfer channel.

There may also be scenarios where the session related data (body) that needs to be conveyed does not directly reside on the endpoint or User Agent. In such scenarios, it is desirable to have a mechanism whereby the SIP message can contain an indirect reference to the desired content. The receiving party would then use this indirect reference to retrieve the content via a non-SIP transfer channel such as HTTP, FTP, or LDAP.

The purpose of content indirection is purely to provide an alternative transport mechanism for SIP MIME body parts. With the exception of the transport mechanism, indirected body parts are equivalent, and should have the same treatment, as in-line body parts.

3. Example Use Cases

There are several example users of such a content indirection mechanism. These are examples only and are not intended to limit the scope or applicability of the mechanism.

3.1 Presence Notification

The information carried in a presence document could potentially exceed the recommended size for a SIP (NOTIFY) request, particularly if the document carries aggregated information from multiple endpoints. In such a situation, it would be desirable to send the NOTIFY request with an indirect pointer to the presence document which could then be retrieved by, for example, HTTP.

Figure 1: Example information flow for presence notification

```
             Watcher                 Presence Server
                |                           |
                |         SUBSCRIBE         |
                |-------------------------->|
                |          200 OK           |
                |<--------------------------|
                |                           |
                |          NOTIFY           |
                |-------------------------->|
                |          200 OK           |
                |<--------------------------|
                |                           |
                |      NOTIFY (w/URI)       |
                |<--------------------------|
                |           200             |
                |-------------------------->|
                |                           |
                |         HTTP GET          |
                |-------------------------->|
                |                           |
                | application/cpim-pidf+xml |
                |<--------------------------|
                |                           |
```

In this example, the presence server returns an HTTP URI pointing to a presence document on the presence server which the watcher can then fetch using an HTTP GET.

3.2 Document Sharing

During an instant messaging conversation, a useful service is document sharing wherein one party sends an IM (MESSAGE request) with an indirect pointer to a document which is meant to be rendered by the remote party. Carrying such a document directly in the MESSAGE request is not appropriate for most documents. Furthermore, the document to be shared may reside on a completely independent server from the originating party.

Figure 2: Example information flow for document sharing

```
               UAC                  UAS        Web Server
                |                    |                |
                |   MESSAGE w/URI    |                |
                |------------------->|                |
                |        200         |                |
                |<-------------------|                |
                |                    |                |
                |                    |    HTTP GET    |
                |                    |--------------->|
                |                    |   image/jpeg   |
                |                    |<---------------|
                |                    |                |
```

In this example, a user wishes to exchange a JPEG image that she has stored on her web server with another user she has a IM conversation with. The JPEG is intended to be rendered inline in the IM conversation. The recepient of the MESSAGE request launches a HTTP GET request to the web server to retrieve the JPEG image.

4. Requirements

It MUST be possible to specify the location of content via a URI [3].

It MUST be possible to specify the disposition of each URI independently.

It MUST be possible to label each URI to identify if and when the content referred to by that URI has changed. Applications of this mechanism may send the same URI more than once. The intention of this requirement is to allow the receiving party to determine if the content referenced by the URI has changed without having to actually retrieve that content. Example ways the URI could be labelled include a sequence number, timestamp, version number, etc.

It MUST be possible to specify the timespan for which a given URI is valid. This may or may not be the same as the lifetime for the content itself.

It MUST be possible for the UAC and the UAS to indicate support of this content indirection mechanism. A fallback mechanism SHOULD be specified in the event that one of the parties is unable to support content indirection.

It MUST be possible for the UAC and UAS to negotiate the type of the indirect content when using the content indirection mechanism.

It MUST be possible for the UAC and UAS to negotiate support for URI scheme(s) to be used in the content indirection mechanism. This is in addition to the ability to negotiate the content type.

It SHOULD be possible to ensure the integrity of the URI when it is received by the remote party.

It MUST be possible to process the content indirection without human intervention.

It MUST allow for indirect transference of content in any SIP message which would otherwise carry that content as a body.

5. Security Considerations

Any content indirection mechanism introduces additional security concerns. By its nature, content indirection requires an extra processing step and information transfer. There are a number of potential abuses of a content indirection mechanism:

> Content indirection allows the initiator to choose an alternative protocol with weaker security or known vulnerabilities for the content transfer. For example, asking the recipient to issue an HTTP request which results in a Basic authentication challenge.
>
> Content indirection allows the initiator to ask the recipient to consume additional resources in the information transfer and content processing, potentially creating an avenue for denial of service attacks. For example, an active FTP URL consuming 2 connections for every indirect content message.
>
> Content indirection could be used as a form of port scanning attack where the indirect content URL is actually a bogus URL pointing to an internal resource of the recipient. The response to the content indirection request could reveal information about open (and vulnerable) ports on these internal resources.
>
> A content indirection URL can disclose sensitive information about the initiator such as an internal user name (as part of an HTTP URL) or possibly geolocation information.

Fortunately, all of these potential threats can be mitigated through careful screening of both the indirect content URIs that are received as well as those that are sent. The clear requirement is that integrity and potentially privacy protection SHOULD be applied to the content indirection URI(s) in a SIP message.

References

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

[2] Rosenberg, J., Schulzrinne, Camarillo, Johnston, Peterson, Sparks, Handley and Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[3] Berners-Lee, Fielding and Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396, August 1996.

Author's Address

Sean Olson
Microsoft
One Microsoft Way
Redmond, WA 98052
US

Phone: +1-425-707-2846
EMail: seanol@microsoft.com
URI: http://www.microsoft.com/rtc

Full Copyright Statement

Acknowledgement

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