SIPPING
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2009

J. Rosenberg
Cisco
J. van Elburg
C. Holmberg
Ericsson
F. Francois
Nortel
S. Schubert (Ed.)
NTT

March 08, 2009

Delivery of Request-URI Targets to User Agents
draft-rosenberg-sip-target-uri-delivery-01

Status of this Memo

Copyright Notice

Abstract

When a Session Initiation Protocol (SIP) proxy receives a request targeted at a URI identifying a user or resource it is responsible for, the proxy translates the URI to a registered or configured contact URI of an agent representing that user or resource. In the process, the original URI is removed from the request. Numerous use cases have arisen which require this information to be delivered to the user agent. This document describes these use cases and defines an extension to the History-Info header field which allows it to be used to support those cases.

Table of Contents

1. Introduction

A key part of the behavior of proxy servers and B2BUA in the Session Initiation Protocol (SIP) [RFC3261] is that they rewrite the Request-URI of requests as the request moves from the User Agent Client (UAC) to the User Agent Server (UAS). This is particularly true for requests outside of a dialog; requests within a dialog have their path dictated primarily by the Route header fields established by the Record-Routes when the dialog was initiated.

The most basic instance of this behavior is the processing executed by the "home proxy" within a domain. The home proxy is the proxy server within a domain which accesses the location information generated by REGISTER messages, and uses it to forward a request towards a UAC. Based on the rules in [RFC3261], when a home proxy receives a SIP request, it looks up the Request-URI in the location database or mapping table, and translates it to the contact(s) that were registered by the UA or configured in the mapping table. This new contact URI replaces the existing Request URI, and causes the request to be forwarded towards the target UA. Consequently, the original contents of the Request URI are lost.

Over the years, this practice of rewriting the Request-URI has proven problematic. Section 4 describes the problems with this mechanism. Section 5 analyzes the architectural issues which drive these problems. Section 6 overviews a mechanism to solve this problem by extending the History-Info header field. Section 7 describes detailed procedures for user agents and proxies.

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. Definitions

3.1. retarget

A Request-URI rewrite operation is considered to be a retargeting operation if the entity to which the request is ultimately delivered could not, based on the policies of the domain of that entity, place the URI prior to translation in the From header field, and have an identity service in its domain sign it. The inverse is not true however. If an entity can legitimately claim the identity prior to the translation operation, it may still be a retargeting. In this

case, it is a matter of domain policy about whether it is or not.

4. Problem Statement

Several problems arise from the practice of rewriting the request URI.

4.1. Unknown Aliases

SIP user agents are associated with an address-of-record (AOR). It is possible for a single UA to actually have multiple AOR associated with it. One common usage for this is aliases. For example, a user might have an AOR of sip:john@example.com but also have the AORs sip:john.smith@example.com and sip:jsmith@example.com. Rather than registering against each of these AORs individually, the user would register against just one of them, and the home proxy would automatically accept incoming calls for any of the aliases, treating them identically and ultimately forwarding them towards the UA. This is common practice in the Internet Multimedia Subsystem (IMS), where it is called implicit registrations and each alias is called a public identity.

It is a common requirement for a UAS, on receipt of a call, to know which of its aliases was used to reach it. This knowledge can be used to choose ringtones to play, determine call treatment, and so on. For example, a user might give out one alias to friends and family only, resulting in a special ring that alerts the user to the importance of the call.

However, based on the procedures in [RFC3261], when an incoming call hits the home proxy, the request URI (which contains the alias) is rewritten to the registered contact(s). Consequently, the alias that was used is lost, and cannot be known to the UAS.

4.2. Unknown GRUU

A variation on the problem in Section 4.1 occurs with Globally Routable User Agent URI (GRUU) [I-D.ietf-sip-gruu]. A GRUU is a URI assigned to a UA instance which has many of the same properties as the AOR, but causes requests to be routed only to that specific instance. It is desirable for a UA to know whether it was reached because a correspondent sent a request to its GRUU or to its AOR. This can be used to drive differing authorization policies on whether the request should be accepted or rejected, for example. However, like the AOR itself, the GRUU is lost in translation at the home proxy. Thus, the UAS cannot know whether it was contacted via the GRUU or its AOR.

4.3. Limited Use Addresses

A limited use address is an SIP URI that is minted on-demand, and passed out to a small number (usually one) remote correspondent. Incoming calls targeted to that limited use address are accepted as long as the UA still desires communications from the remote target. Should they no longer wish to be bothered by that remote correspondent, the URI is invalidated so that future requests targeted to it are rejected.

Limited use addresses are used in battling voice spam [RFC5039]. The easiest way to provide them would be for a UA to be able to take its AOR, and "mint" a limited use address by appending additional parameters to the URI. It could then give out the URI to a particular correspondent, and remember that URI locally. When an incoming call arrives, the UAS would examine the parameter in the URI and determine whether or not the call should be accepted. Alternatively, the UA could push authorization rules into the network, so that it need not even see incoming requests that are to be rejected.

This approach, especially when executed on the UA, requires that

parameters attached to the AOR, but not used by the home proxy in processing the request, will survive the translation at the home proxy and be presented to the UA. This will not be the case with the logic in RFC 3261, since the Request-URI is replaced by the registered contact, and any such parameters are lost.

4.4. Sub-Addressing

Sub-Addressing is very similar to limited use addresses. Sub-addresses are addresses within a subdomain that are multiplexed into a single address within a parent domain. The concept is best illustrated by example. Consider a VoIP service provided to consumers. A consumer obtains a single address from its provider, say sip:family@example.com. However, Joe is the patriarch of a family with four members, and would like to be able to have a separate identifier for each member of his family. One way to do that, without requiring Joe to purchase new addresses for each member from the provider, is for Joe to mint additional URI by adding a parameter to the AOR. For example, his wife Judy with have the URI sip:family@example.com;member=judy, and Joe himself would have the URI sip:family@example.com;member=joe. The SIP server provider would receive requests to these URI, and ignoring the unknown parameters (as required by [RFC3261]) route the request to the registered contact, which corresponds to a SIP server in Joes home. That server, in turn, can examine the URI parameters and determine which phone in the home to route the call to.

This feature is not specific to VoIP, and has existing in Integrated Services Digital Networking (ISDN) for some time. It is particularly useful for small enterprises, in addition to families. It is also similar in spirit (though not mechanism) to the ubiquitous home routers used by consumers, which allow multiple computers in the home to "hide" behind the single IP address provided by the service provider, by using the TCP and UDP port as a sub-address.

The sub-addressing feature is not currently feasible in SIP because of the fact that any SIP URI parameter used to convey the sub-address would be lost at the home proxy, due to the fact that the Request-URI is rewritten there.

4.5. Service Invocation

Several SIP specifications have been developed which make use of complex URIs to address services within the network rather than subscribers. The URIs are complex because they contain numerous parameters that control the behavior of the service. Examples of this include the specification which first introduced the concept, [RFC3087], control of network announcements and IVR with SIP URI [RFC4240], and control of voicemail access with SIP URI [RFC4458].

A common problem with all of these mechanisms is that once a proxy has decided to rewrite the Request-URI to point to the service, it cannot be sure that the Request-URI will not be destroyed by a downstream proxy which decides to forward the request in some way, and does so by rewriting the Request-URI.

4.6. Freephone Numbers

Freephone numbers, also known as 800 or 8xx numbers in the United States, are telephone numbers that are free for users to call (although the author will note that such notions are becoming less important as billing models evolve, and harken back to an era where phone service depended on global agreement on such billing concepts).

In the telephone network, freephone numbers are just aliases to actual numbers which are used for routing of the call. In order to process the call in the PSTN, a switch will perform a query (using a protocol called TCAP), which will return either a phone number or the identity of a carrier which can handle the call.

There has been recent work on allowing such PSTN translation services to be accessed by SIP proxy servers through IP querying mechanisms. ENUM, for example [RFC3761] has already been proposed as a mechanism for performing Local Number Portability (LNP) queries [RFC4769], and recently been proposed for performing calling name queries

[I-D.ietf-enum-cnam]. Using it for 8xx number translations is a logical next-step.

Once such a translation has been performed, the call needs to be routed towards the target of the request. Normally, this would happen by selecting a PSTN gateway which is a good route towards the translated number. However, one can imagine all-IP systems where the 8xx numbers are SIP endpoints on an IP network, in which case the

translation of the 8xx number would actually be a SIP URI and not a phone number. Assuming for the moment it is a PSTN connected entity, the call would be routed towards a PSTN gateway. Proper treatment of the call in the PSTN (and in particular, correct reconciliation of billing records) requires that the call be marked with both the original 8xx number AND the target number for the call. However, in our example here, since the translation was performed by a SIP proxy upstream from the gateway, the original 8xx number would have been lost, and the call will not interwork properly with the PSTN.

Similar problems arise with other "special" numbers and services used in the PSTN, such as operator services, pay numbers (9xx numbers in the U.S), and short service codes such as 311.

5. Architectural Roots of the Problem

There is a common theme across all of the problems in Section 4, and this theme is the confounding of names, routes, and addresses.

A name is a moniker for an entity which refers to it in a way which reveals nothing about where it is in a network. In SIP, tel URI which doesn't represent the location of the entity is a name. An address is an identifier for an entity which describes it by its location on the network. In SIP, the SIP URI itself is a form of address because the host part of the URI, the only mandatory part of the URI besides the scheme itself, indicates the location of a SIP server that can be used to handle the request. Finally, a route is a sequence of SIP entities (including the UA itself!) which are traversed in order to forward a request to an address or name.

SIP, unfortunately, uses the Request-URI as a mechanism for routing of the request in addition to using it as the mechanism for identifying the name or address to which the request was targeted. A home proxy rewrites the Request-URI because that rewriting is the vehicle by which the request is forwarded to the target of the request. However, this rewritten URI (the contact from the register), is not in any way a meaningful name or address for the UA. Indeed, with specifications like SIP outbound [I-D.ietf-sip-outbound], even the IP address within the registered

contact is meaningless since the flow on which the REGISTER is sent

is used rather than the IP address. Consequently, the home proxy is fundamentally replacing the *address* in the Request-URI with a *route* to reach that UA. This architectural mistake is the cause of the problems described above.

Interestingly, this same mistake was present in [RFC2543] for the handling of mid-dialog requests. It was fixed through the loose routing mechanism in RFC 3261, which used Route header fields to identify each hop to visit for a mid-dialog request, and separated this from the Request-URI, which identified the ultimate target of the request (the remote UA), and remained unmodified in the processing of the request.

Unfortunately, application of this same technique to address the problem at hand cannot be done in a backwards compatible manner. Consequently, some other means is needed to clearly identify which URIs are addresses, and which are routes. To avoid confusion, we refer to a SIP URI that is an address for a user or resource as a "target" and a SIP URI that is a hop for reaching that user as a "hop".

6. Solution Overview

The History-Info header field, defined in [RFC4244], defines a mechanism by which an enumeration of the URIs traversed can be passed to both the UAC and UAS. History-Info was designed to provide a general purpose mechanism which can support the needs of many applications, including diagnostics and traditional telephony features like voicemail. Were a home proxy to implement History-Info, it would provide a means for that proxy to deliver the target URI to the UAS.

Unfortunately, History-Info makes no distinction between URIs that are targets and URIs that are hops. Consequently, if there were additional proxies downstream of the home proxy which modified the Request-URI in any way, the UA would have no way to know which URI in the list of History-Info values was actually the target. To remedy that, this document defines an extension to the History-Info header field which indicates whether the URI is a target or not.

When a home proxy receives a request for a user or resource for which it has a registration, the proxy adds two History-Info header field values. The first is the incoming request URI. Since the Request-URI identifies a user or resource for which it has a registration, the Request-URI is an AOR and thus an address for the user. The proxy adds a History-Info header field parameter, "istarget", which

indicates this. Next, the proxy inserts the contact URI it used in the outgoing Request-URI. No "istarget" parameter is included in this History-Info value.

For a UA to determine the URI target, it need only walk backwards through the list of HI values, and take the first one containing the "istarget" parameter.

For example, consider the architecture in Figure 1. In the example user A calls user B. User B is in example.com. The call from A to B passes through A's outbound proxy, their home proxy, B's home proxy, and B's outbound proxy, prior to reaching B. B's home proxy, H-B, performs the translation of the R-URI to the registered contact based on the registration database. Consequently, it adds two History-Info header fields, the first of which represents the incoming R-URI and includes the "istarget" parameter.

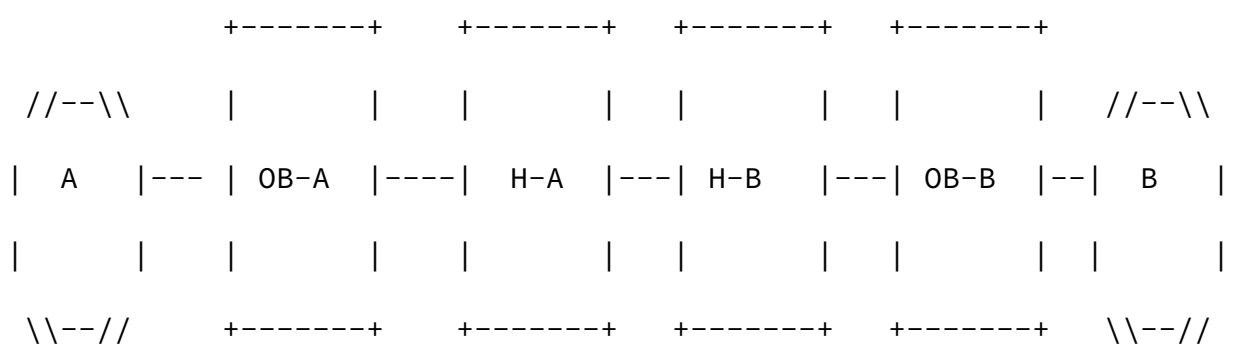
A
OB-A
H-A
H-B
OB-B
B
INVITE
sip:B@example.com
INVITE
sip:B@example.com
INVITE
sip:B@example.com
INVITE
sip:B@example.com
HI: <sip:B@example.com>index=1;istarget,
<sip:B@1.2.3.4>;index=1.1

Figure 1: Target URI Example

7. Detailed Semantics

The "istarget" parameter in the History-Info header field indicates that the URI that it parameterizes was either subject to a lookup in a location service created through the registration process of the UA or was available through configured mapping. Furthermore, if that URI had an 'index' of N, the URIs with indices N.M for all M, are the registered contacts to that URI.

7.1. Proxy Behavior

A proxy compliant to this specification SHOULD add a History-Info header field value to a request under the following conditions:

- The proxy is responsible for the domain in AOR in the Request-URI
- The proxy will be translating the contents of the Request-URI to one or more contacts either based on a location database populated through REGISTER requests from user agents or based on configured mapping.
- The R-URI exists in the location database.

The proxy SHOULD populate the History-Info header field regardless of whether there is a Supported header field with value 'histinfo'. If the incoming request already contains a History-Info header field, and the last value of that header field is identical to the Request-URI of the received request, the proxy MUST add a "istarget" attribute to that History-Info value. If the request did not contain a History-Info header field, or if it did, but the last value is not identical to the Request-URI of the received request, the proxy MUST add another History-Info header field value. The URI MUST be equal

to the incoming Request-URI, and MUST contain a "istarget" attribute. The index is set as defined in [RFC4244].

Once the proxy has translated the Request-URI into a registered contact or configured contact, it MUST add an additional History-Info header field value containing the Contact URI for each request to be forwarded. The "istarget" attribute MUST NOT be present. The index is set as defined in [RFC4244].

Since the principal purpose of the "istarget" parameter is to indicate, to a UAS, the target URI by which it was reached, there is no need for the History-Info header field values to be passed outside of the domain which inserted them, unless there is an apparent need for passing on the value downstream (e.g. freephone number).

If the proxy is actually redirecting and not forwarding the request,

it SHOULD include a History-Info URI in the response for the target. That URI, if present, MUST contain the "istarget" attribute. It SHOULD NOT add a History-Info URI for the registered contact; the previous hop proxy will do that. Note that, this rule violates a SHOULD-strength rule in Section 4.3.4 of [RFC4244]. That section indicates that redirections "SHOULD NOT" contain any new History-Info header fields, as those will be added by the upstream server. For this application however, only the downstream server knows that the R-URI was a target, and thus the History-Info header field and the "istarget" attribute must be added by the downstream server.

7.2. UA Behavior

A UAS receiving a request, and wishing to determine the original target dialog, takes the values in the History-Info header field, and traverses through them in reverse order. Note that, the value of the "index" attribute is not relevant; the traversal is in order of the header fields values themselves. The UAS finds the first header field value containing the "istarget" parameter. If such a value does not exist, the target URI cannot be reliably determined. If it does exist, the URI is examined. If the domain of the URI matches the domain of the UA, based on the UA's configured awareness of its own domain, that URI is the target URI. If the domains do not match, the target URI cannot be reliably determined. This domain check is present to handle cases where a request is forwarded through two

separate domains, and the domain of the actual UAS didn't support this specification, but the previous domain did. If there are more than one header field value containing "istarget" parameter, handling of second and latter value with "istarget" parameter is up to local policy and is outside the scope of this document. For example, if freephone number was invoked, there may be two header field value with "istarget" parameter;one indicating the retargeting of freepone number to a corporate address and another indicating the retargeting of corporate address to a registered contact.

NOTE: Do we want to introduce another parameter to indicated the difference between retargeting based on location lookup and configured mapping?

Beyond this, there is no special UA processing associated with the "istarget" parameter.

8. The difference to P-Called-Party-Id

As defined in [RFC3455], if a SIP entity, which acts as registrar/ home proxy for the terminating user, re-writes the Request-URI with the contact address of the registered UA it may insert a P-Called-

Party-ID header field with the previous value of the Request-URI.

The last hi-entry in History-Info minted with an "istarget" attribute and P-Called-Party-ID header field have different semantics. The last hi-entry in History-Info minted with an "istarget" attribute represents the current target identity, while the P-Called-Party-ID represents the last Request-URI value used to reach the user before the Request-URI value was re-written with the Contact address of the UAS. In some cases the P-Called-Party-ID may be the same as the current target but, it may also be the last route taken (not equal to the current target) to deliver the request. Therefore the P-Called-Party-ID can not be used in a generic SIP environment to represent the current target.

3GPP has defined procedures for the usage of P-Called-Party-ID, so 3GPP would need to continue to use the header, in addition to the new Target header. However, both mechanisms can exist in parallel.

9. Syntax

This specification extends the syntax of hi-param in Section 4.1 of RFC 4244:

```
hi-param = hi-index / hi-target / hi-extension

hi-target = "istarget"
```

10. Security Considerations

The "istarget" parameter indicates that a URI was subject to translation by a home proxy, and consequently, acts as an explicit indicator that a particular URI was an AOR for a user. This might be useful for attackers wishing to farm requests for targettable URIs for purposes of spamming. Of course, such attackers can utilize URIs in History-Info even if they lack the "istarget" attribute, so "istarget" does not really exacerbate this. Nonetheless, since the princpal application of the "istarget" parameter is delivery of a URI to a UAS within the same domain, History-Info values inserted solely for this purpose SHOULD be removed at the domain boundary.

11. References

11.1. Normative References

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

[RFC4244] Barnes, M., "An Extension to the Session Initiation

Protocol (SIP) for Request History Information", RFC 4244, November 2005.

11.2. Informative References

[I-D.ietf-sip-gruu]
Rosenberg, J., "Obtaining and Using Globally Routable User Agent (UA) URIs (GRUU) in the Session Initiation Protocol (SIP)", draft-ietf-sip-gruu-15 (work in progress), October 2007.

[RFC5039] Rosenberg, J. and C. Jennings, "The Session Initiation Protocol (SIP) and Spam", RFC 5039, January 2008.

[RFC3087] Campbell, B. and R. Sparks, "Control of Service Context using SIP Request-URI", RFC 3087, April 2001.

[RFC4240] Burger, E., Van Dyke, J., and A. Spitzer, "Basic Network Media Services with SIP", RFC 4240, December 2005.

[RFC4458] Jennings, C., Audet, F., and J. Elwell, "Session Initiation Protocol (SIP) URIs for Applications such as Voicemail and Interactive Voice Response (IVR)", RFC 4458, April 2006.

[RFC2543] Handley, M., Schulzrinne, H., Schooler, E., and J. Rosenberg, "SIP: Session Initiation Protocol", RFC 2543, March 1999.

[RFC3455] Garcia-Martin, M., Henrikson, E., and D. Mills, "Private Header (P-Header) Extensions to the Session Initiation Protocol (SIP) for the 3rd-Generation Partnership Project (3GPP)", RFC 3455, January 2003.

[RFC3761] Faltstrom, P. and M. Mealling, "The E.164 to Uniform Resource Identifiers (URI) Dynamic Delegation Discovery System (DDDS) Application (ENUM)", RFC 3761, April 2004.

[RFC4769] Livingood, J. and R. Shockey, "IANA Registration for an Enumservice Containing Public Switched Telephone Network (PSTN) Signaling Information", RFC 4769, November 2006.

[I-D.ietf-sip-outbound]
Jennings, C. and R. Mahy, "Managing Client Initiated Connections in the Session Initiation Protocol (SIP)", draft-ietf-sip-outbound-16 (work in progress), October 2008.

[I-D.ietf-enum-cnam]
Shockey, R., "IANA Registration for an Enumservice Calling Name Delivery (CNAM) Information and IANA Registration for URI type 'pstndata'", draft-ietf-enum-cnam-08 (work in progress), September 2008.

Authors' Addresses

Jonathan Rosenberg
Cisco
Edison, NJ
US

Email: jdrosen@cisco.com
URI: http://www.jdrosen.net

Hans Erik van Elburg
Ericsson
Ericssonstraat 2
Rijen 5121 ML
The Netherlands

Email: HansErik.van.Elburg@ericsson.com

Christer Holmberg
Ericsson
Hirsalantie 11, Jorvas
Finland

Email: christer.holmberg@ericsson.com

Francois Audet
Nortel

Email: audet@nortel.com

Shida Schubert (editor)
NTT

Email: shida at ntt-at.com