Network Working Group
Internet-Draft
Obsoletes: 4244 (if approved)
Intended status: Standards Track
Expires: December 26, 2010

M. Barnes
Polycom
F. Audet
Skype
S. Schubert
NTT
J. van Elburg
Detecon International Gmbh
C. Holmberg
Ericsson

June 24, 2010

An Extension to the Session Initiation Protocol (SIP) for Request History Information
draft-ietf-sipcore-rfc4244bis-01.txt

Abstract

This document defines a standard mechanism for capturing the history information associated with a Session Initiation Protocol (SIP) request. This capability enables many enhanced services by providing the information as to how and why a call arrives at a specific application or user. This document defines an optional SIP header, History-Info, for capturing the history information in requests. A SIP/SIPS URI parameter is defined to tag information necessary to populate the History-Info header. In addition, this document defines a value for the Privacy header specific to the History-Info header.

Table of Contents

1. Introduction 4
2. Conventions and Terminology 4
3. Overview of Operations 5
4. General User Agent Behavior 9
4.1. User Agent Client (UAC) Behavior 9
4.2. User Agent Server (UAS) Behavior 10
4.2.1. Processing of Requests with History-Info 10
4.2.2. Generation of Responses with History-Info 10
4.3. Redirect Server Behavior 11
5. Proxy Behavior 11
5.1. Adding the History-Info Header to Requests 11
5.1.1. Initial Request 11
5.1.2. Re-sending based on failure response 12
5.1.3. Re-sending based on redirection response 13
5.2. Sending History-Info in Responses 14
6. The History-Info header field 14
6.1. Definition 14
6.2. Examples 16
6.3. Procedures 16
6.3.1. SIP/SIPS URI target parameter for History-Info Header 17
6.3.2. Privacy in the History-Info Header 17
6.3.3. Reason in the History-Info Header 18
6.3.4. Indexing in the History-Info Header 19
6.3.5. Request Target in the History-Info Header 20
7. Application Considerations 20
8. Security Considerations 22
9. IANA Considerations 22
9.1. Registration of New SIP History-Info Header 22
9.2. Registration of "history" for SIP Privacy Header 23
9.3. Registration of "hit" SIP/SIPS URI Parameter 23
10. Acknowledgements 24
11. Changes from RFC 4244 24
11.1. Backwards compatibility 26
12. Changes since last Version 26
13. References 29
13.1. Normative References 29
13.2. Informative References 30
Appendix A. Request History Requirements 30
A.1. Security Requirements 32
A.2. Privacy Requirements 32
Appendix B. Example call flows 33
B.1. Sequentially Forking (History-Info in Response) 33
B.2. History-Info with Privacy Header 40
B.3. Privacy Header for a Specific History-Info Entry 41
Authors' Addresses 43

1. Introduction

Many services that SIP is anticipated to support require the ability to determine why and how the call arrived at a specific application. Examples of such services include (but are not limited to) sessions initiated to call centers via "click to talk" SIP Uniform Resource Locators (URLs) on a web page, "call history/logging" style services within intelligent "call management" software for SIP User Agents (UAs), and calls to voicemail servers. Although SIP implicitly provides the redirect/retarget capabilities that enable calls to be routed to chosen applications, there is a need for a standard mechanism within SIP for communicating the retargeting history of such a request. This "request history" information allows the receiving application to determine hints about how and why the call arrived at the application/user.

This document defines a SIP header, History-Info, to provide a standard mechanism for capturing the request history information to enable a wide variety of services for networks and end-users. A SIP/SIPS URI parameter is defined to tag information necessary to populate the History-Info header. In addition, this document defines a value for the Privacy header specific to the History-Info header.

The History-Info header provides a building block for development of SIP based applications and services. The requirements for the solution described in this document are included in Appendix A. Example scenarios using the History-Info header are included in Appendix B.

2. Conventions and 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].

The term "retarget" is used in this document to refer both to the process of a Proxy Server/User Agent Client (UAC) changing a Uniform Resource Identifier (URI) in a request based on the rules for determining request targets as described in Section 16.5 of [RFC3261] and the subsequent forwarding of that request as described in section 16.6 of [RFC3261].

The term "forward" is used consistent with the terminology in [RFC3261]. Noting that [RFC3261] uses the term "forwarding" to describe a proxy's handling of requests for domains for which is not responsible, as well as to describe the basic "forwarding" of a request (in section 16.6) once a target has been determined.

However, the context of the usage is sufficient to differentiate the slightly different meanings.

The terms "location service", "redirect" and "AOR" are used consistent with the terminology in [RFC3261].

3. Overview of Operations

SIP implicitly provides retargeting capabilities that enable calls to be routed to specific applications as defined in [RFC3261]. The motivation for capturing the request history is that in the process of retargeting a request, old routing information can be forever lost. This lost information may be important history that allows elements to which the call is retargeted to process the call in a locally defined, application-specific manner. This document defines a mechanism for transporting the request history. Application-specific behavior is outside the scope of this specification.

Current network applications provide the ability for elements involved with the call to exchange additional information relating to how and why the call was routed to a particular destination. The following are examples of such applications:

1. Web "referral" applications, whereby an application residing within a web server determines that a visitor to a website has arrived at the site via an "associate" site that will receive some "referral" commission for generating this traffic

2. Email forwarding whereby the forwarded-to user obtains a "history" of who sent the email to whom and at what time

3. Traditional telephony services such as voicemail, call-center "automatic call distribution", and "follow-me" style services

Several of the aforementioned applications currently define application-specific mechanisms through which it is possible to obtain the necessary history information.

In addition, request history information could be used to enhance basic SIP functionality by providing the following:

- Some diagnostic information for debugging SIP requests.

- Capturing aliases and Globally Routable User Agent URIs (GRUUs) [RFC5627], which can be overwritten by a home proxy upon receipt of the initial request.

- Facilitating the use of limited use addresses (minted on demand) and sub-addressing.

- Preserving service specific URIs that can be overwritten by a downstream proxy, such as those defined in [RFC3087], and control of network announcements and IVR with SIP URI [RFC4240].

- A stronger security solution for SIP. A side effect is that each proxy that captures the "request history" information in a secure manner provides an additional means (without requiring signed keys) for the original requestor to be assured that the request was properly retargeted.

The fundamental functionality provided by the request history information is the ability to inform proxies and UAs involved in processing a request about the history or progress of that request (CAPABILITY-req, see Appendix A). The solution is to capture the Request-URIs as a request is retargeted, in a SIP header: History-Info (CONTENT-req, see Appendix A). This allows for the capturing of the history of a request that would be lost with the normal SIP processing involved in the subsequent retargeting of the request.

The History-Info header can appear in any request not associated with an established dialog (e.g., INVITE, REGISTER, MESSAGE, REFER and OPTIONS, PUBLISH and SUBSCRIBE, etc.) (REQUEST-VALIDITY-req, see Appendix A) and any valid response to these requests (ISSUER-req, see Appendix A).

This specification defines parameters detailed in Section 6.1 for carrying the following information in the History-Info header:

- Targeted-to-URI: The targeted-to-URI entry captures the Request-URI for the specific request as it is forwarded.

- Index: The index reflects the chronological order of the information, indexed to also reflect the forking and nesting of requests.

- Reason: Reason describes why an entry was retargeted.

- Privacy: Privacy is used to request that an entry be anonymized in the case of a request that is retargeted to a domain for which the retargeting entity is not responsible.

- Target: The target parameter indicates whether the Targeted-to-URI is a registered contact ("rc") for or another user mapped ("mp") from the Request-URI in the incoming request that was retargeted.

This specification also defines the "hit" SIP/SIPS URI Parameter. This parameter is added to URIs as they are added to the target list for a request based on the same criteria as the target parameter described above. This URI parameter is used to determine the value for the target parameter. Further detail is provided in Section 6.3.1.

In addition, this specification defines a value for the Privacy header, "history", that applies to all the History-Info header entries in a Request or to a specific History-Info header entry as described above. Further detailed is provided in Section 6.3.2.

The following is an illustrative example of usage of History-Info.

In this example, Alice (sip:alice@atlanta.example.com) calls Bob (sip:bob@biloxi.example.com). Alice's home proxy (sip: atlanta.example.com) forwards the request to Bob's proxy (sip: biloxi.example.com). When the request arrives at sip: biloxi.example.com, it does a location service lookup for bob@biloxi.example.com and changes the target of the request to Bob's Contact URIs provided as part of normal SIP registration. In this example, Bob is simultaneously contacted on a PC client and on a phone, and Bob answers on the PC client.

One important thing illustrated by this call flow is that without History-Info, Bob would "lose" the target information, including any parameters in the request URI. Bob can now recover that information by locating the first entry just prior to the last hi-entry marked as "rc", which is one level up in indexing from the last hi-entry - i.e., the sip:bob@biloxi.example.com entry with index=1.1.

The formatting in this scenario is for visual purposes; thus, backslash and CRLF are used between the fields for readability and the headers in the URI are not shown properly formatted for escaping. Refer to Section 6.2 for the proper formatting. Additional detailed scenarios are available in the Appendix B.

Note: This example uses loose routing procedures.

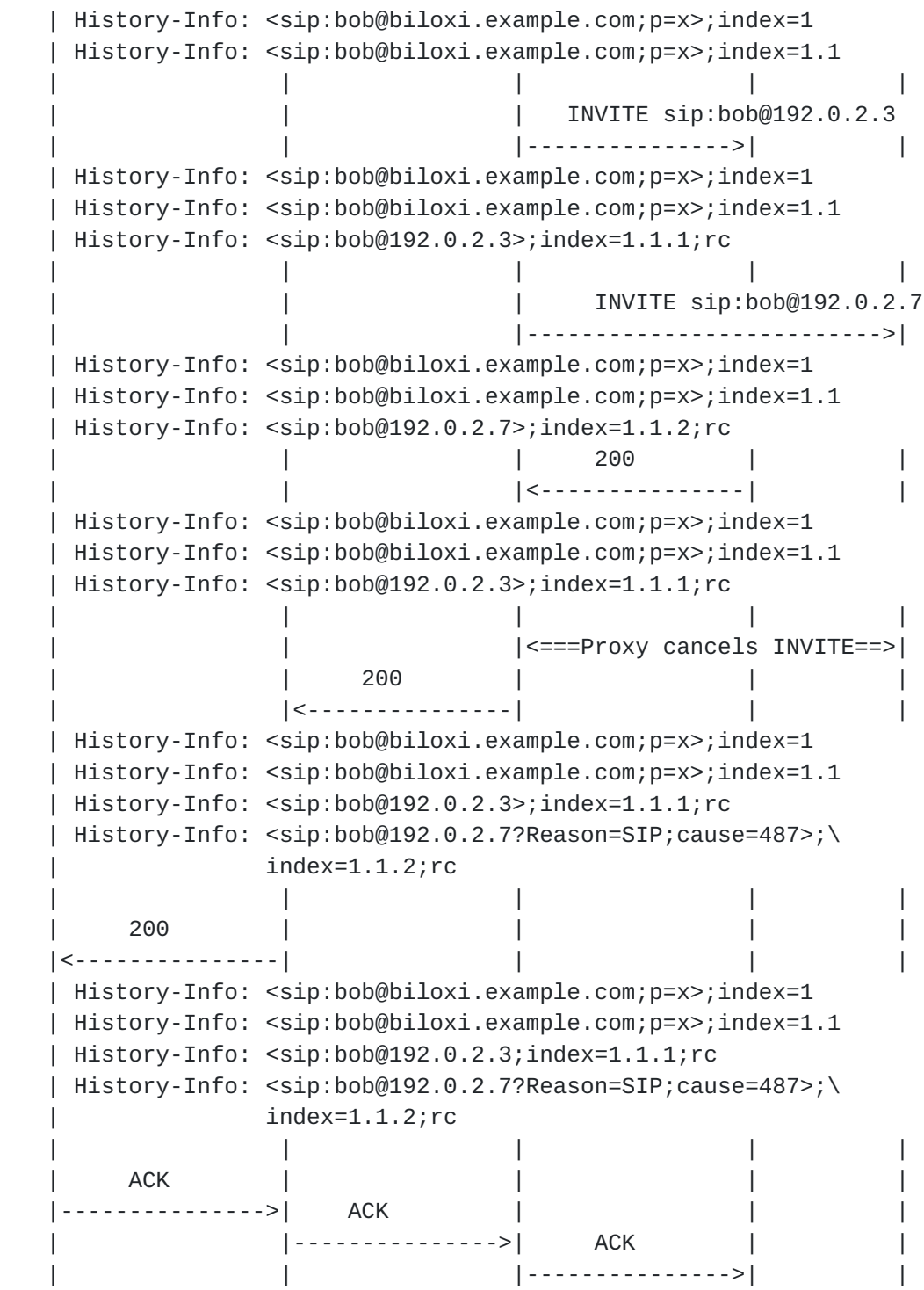


Figure 1: Basic Call

4. General User Agent Behavior

This section describes the processing specific to UAs for the History-Info header.

4.1. User Agent Client (UAC) Behavior

The UAC SHOULD include the "histinfo" option tag in the Supported header in any request not associated with an established dialog for which the UAC would like the History-Info header in the response. In addition, the UAC MAY add a History-Info header, using the Request-URI of the request as the hi-target-to-uri, in which case the index MUST be set to a value of 1 in the hi-entry. As a result, intermediaries and the UAS at least know the original Request-URI, and if the Request-URI was modified by a previous hop. Normally, UACs are not expected to include a History-Info header in an initial request as it is more of a Proxy function; the main reason it is allowed is for B2BUAs who are performing proxy-like functions such as routing.

A UAC that does not want an hi-entry added due to privacy considerations MUST include a Privacy header with a priv-value(s) of "header" or "history." A UAC that wants to ensure that privacy not be applied to its identity MUST include a Privacy header with a priv-value of "none."

In the case where a UAC receives a 3xx response with a Contact header and sends a new request in response to it, the UAC MAY include in the outgoing request the previous hi-entry(s) received in the response. In this case, the reason header MUST be associated with the hi-targeted-to-uri in the previous (last) hi-entry, as described in Section 6.3.3. A new hi-entry MUST then be added for the URI from the Contact header (which becomes the new Request-URI). An index MUST be added to the hi-entry. The value for the index is determined by reading and incrementing the value of the index from the previous hi-entry, thus following the same rules as those prescribed for a proxy in retargeting, described in Section 6.3.4. If the URI in the Contact header contains a "hit" URI parameter, the UAC MUST add a target parameter to the hi-entry and MUST remove the URI parameter as described in Section 6.3.5.

Prior to any application usage of the information by the UAC (e.g., debug), the validity SHOULD be ascertained. The entries SHOULD be evaluated to determine gaps in indices, which could indicate that an entry has been maliciously removed. Either way, an application SHOULD be aware of potentially missing information. The interpretation of the information in the History-Info header by a UAC in a request depends upon the specific applications supported by the

UAC. Application considerations and guidelines are provided in section 7.

4.2. User Agent Server (UAS) Behavior

4.2.1. Processing of Requests with History-Info

Once the request terminates at the UAS, the UAS evaluates the History-Info header. The last hi-entry reflects the most recent target and SHOULD contain the Request-URI for the received request. If the Request-URI of the incoming request does not match the last hi-entry (e.g., the last proxy does not support History-Info), the UAS MUST insert an hi-entry. The UAS MUST set the hi-targeted-to-uri based to the value of Request-URI in the incoming request, unless privacy is required. If privacy is required, the procedures of Section 6.3.2 MUST be used. The UAS MUST include an hi-index attribute as described in Section 6.3.4. The UAS MUST NOT include a hi-target attribute, since the UAS has no way to know the mechanism by which the Request-URI was determined. The addition of the missing hi-entry ensures that the most complete information can be provided in the response and provides consistency in the information presented to applications. The information can also be useful for implementations with B2BUAs that include the History-Info, received in the incoming request, in the subsequent outgoing request.

Prior to any application usage of the information, the validity SHOULD be ascertained. The entries SHOULD be evaluated to determine gaps in indices, which could indicate that an entry has been maliciously removed. Either way, an application SHOULD be aware of potentially missing information. The interpretation of the information in the History-Info header by a UAS in a request depends upon the specific applications supported by the UAS. Application considerations and guidelines are provided in section 7.

4.2.2. Generation of Responses with History-Info

If the "histinfo" option tag is received in a request, the UAS MUST include any History-Info received in the request in the subsequent response. If privacy is required, entries MUST be anonymized using [RFC3323]. The UAS MUST follow the rules for a redirect server per Section 4.3 in generating a 3xx response.

The processing of History-Info in responses follows the methodology described in Section 16.7 of [RFC3261], with the processing of History-Info headers adding an additional step, just before Step 9, "Forwarding the Response".

4.3. Redirect Server Behavior

A redirect server MUST include the History-Info headers received in the request in the 3XX response that it sends. A redirect server MUST NOT add any new History-Info entries. In order to provide the necessary information to populate the target parameter for the new History-Info entries as a request is redirected, the redirect server MUST add a "hit" URI parameter to the URIs in the contact header. The appropriate values for this parameter are described in Section 6.3.1.

5. Proxy Behavior

The specific processing by proxies for adding the History-Info headers in SIP requests and responses is described in this section.

5.1. Adding the History-Info Header to Requests

This section describes the process of adding the History-Info header to requests for the following cases:

- Forwarding of initial request (see Section 5.1.1)
- Resending based on failure response (see Section 5.1.2)
- Resending based on redirection response (see Section 5.1.3)

Retargeting is an iterative process, i.e., a proxy may redirect "internally " more than one time. A typical example would be a proxy that redirects a request first to a different user (i.e., it maps to a different AOR), and then forwards to a registered contact bound to that new AOR. A proxy that uses such mechanism SHOULD add multiple hi-entry fields (e.g., bob@example.com to office@example.com to office@192.0.2.5) to provide a logical description of the retargeting process. A Reason MAY be associated with the hi-targeted-to-uri that has been retargeted as shown in the example in Appendix B.1.

5.1.1. Initial Request

Upon receipt of an initial request for a dialog, or a standalone request, a proxy forwarding the request MUST perform the following steps. Note that those steps below do not apply if the request is being re-sent as a result of failure (i.e., timeout, reception of an error response), or redirection caused by receipt of a 3XX message).

Step 1: Adding Entries on Behalf of Previous Hops

If an incoming request does not already have a History-Info header field (e.g., the UAC does not include any History-Info header and no proxies in between support History-Info), or if the Request-URI of the incoming request does not match the last hi-entry (e.g., the last proxy does not support History-Info), the proxy MUST insert an hi-entry. The proxy MUST set the hi-targeted-to-uri based to the value of Request-URI in the incoming request, unless privacy is required. If privacy is required, the procedures of Section 6.3.2 MUST be used. The proxy MUST NOT include a hi-target attribute. The proxy MUST include an hi-index attribute as described in Section 6.3.4.

Step 2: Tagging URIs in Target Set with "hit" Parameter

The proxy then proceeds to determining the request targets as per 16.5/[RFC3261]. The proxy MUST add a "hit" SIP/SIPS URI parameter for the target URI that are determined based on either of the two mechanisms as described in Section 6.3.1.

Step 3: Generating New Entries for Each Outgoing Request

The proxy then proceeds to request forwarding as per 16.6/ [RFC3261]. The proxy MUST add a separate hi-entry in each separate outgoing request for each of the current (outgoing) targets in the target set. The proxy MUST set the hi-targeted-to-uri in those separate hi-entry(s) to the value of the Request-URI of the current (outgoing) request, unless privacy is required. If privacy is required, the procedures of Section 6.3.2 MUST be used. The proxy MUST include an hi-index for each of the separate hi-entry(s) as described in Section 6.3.4. If the Request-URI contains a "hit" SIP/SIPS URI parameter, the proxy MUST include a hi-target attribute for each of the separate entry(s) as described in Section 6.3.5. The proxy MUST remove the "hit" URI parameter from the Request-URI.

5.1.2. Re-sending based on failure response

When re-sending a request as a result of retargeting because of failure (i.e., either reception of error responses or a timeout which is considered to be an implicit 487 error response), the proxy MUST perform the following steps:

Step 1: Including the Entries from Error Responses & Timeouts

The proxy MUST build the History-Info header field(s) sent in the outgoing request using the aggregate information associated with the received error responses(s) and timeout(s) for all the branches that are generating failures, including the header entries in the order indicated by the indexing (see Section 6.3.4). If the received error response did not include any History-Info header fields, the proxy MUST use the same History-Info header fields that were sent in the outgoing request that failed to build the outgoing request.

Step 2: Tagging the Last Entries

The proxy then examines the last hi-entry of the History-Info that was just generated in Step 1 for each one of the branches that generated failures or timeouts and MUST add a Reason header for each one of those entries as per the procedures of Section 6.3.3.

Step 3: Generating New Entries for Each Outgoing Requests

Same as per Step 3 above for the normal forwarding case Section 5.1.1.

5.1.3. Re-sending based on redirection response

When re-sending a request as a result of retargeting because of redirection (i.e., receipt of a 3XX response), the following steps apply:

Step 1: Including Previous Entries

The proxy MUST include the History-Info header fields that were sent in the outgoing request that is being redirected.

Step 2: Tagging the Last Entry

The proxy then examines the last hi-entry of the History-Info that was just generated in Step 1 and MUST add a Reason header this entry as per the procedures of Section 6.3.3.

Step 3: Generating New Entries for Each Outgoing Requests

Same as per Step 3 for the normal forwarding case Section 5.1.1.

5.2. Sending History-Info in Responses

A proxy that receives a request with the "histinfo" option tag in the Supported header, SHOULD forward captured History-Info in subsequent, provisional, and final responses to the request sent by the ultimate UAS (see Section 4.2).

A proxy MAY anonymize any hi-entry whose domain corresponds to a domain for which it is responsible (as per [RFC3323]). For example, a proxy could require that such entries be anonymized when a response is forwarded to a domain for which the proxy is not responsible.

The processing of History-Info in responses follows the methodology described in Section 16.7 of [RFC3261], with the processing of History-Info headers adding an additional step, just before Step 9, "Forwarding the Response".

6. The History-Info header field

6.1. Definition

History-Info is a header field as defined by [RFC3261]. "It may appear in any initial request for a dialog, standalone request or responses associated with these requests. For example, History-Info may appear in INVITE, REGISTER, MESSAGE, REFER, OPTIONS, SUBSCRIBE, and PUBLISH and any valid responses, plus NOTIFY requests that initiate a dialog.

The History-Info header carries the following information, with the mandatory parameters required when the header is included in a request or response:

- Targeted-to-URI (hi-targeted-to-uri): A mandatory parameter for capturing the Request-URI for the specific request as it is forwarded.

- Index (hi-index): A mandatory parameter for History-Info reflecting the chronological order of the information, indexed to also reflect the forking and nesting of requests. The format for this parameter is a string of digits, separated by dots to indicate the number of forward hops and retargets. This results in a tree representation of the history of the request, with the lowest-level index reflecting a branch of the tree. By adding the new entries in order (i.e., following existing entries per the details in Section 5.1), including the index and securing the header, the ordering of the History-Info headers in the request is assured (SEC-req-2, see Appendix A.1). In addition, applications

may extract a variety of metrics (total number of retargets, total number of retargets from a specific branch, etc.) based upon the index values.

- Reason: An optional parameter for History-Info, reflected in the History-Info header by including the Reason Header [RFC3326] escaped in the hi-targeted-to-uri. A reason is included for the hi-targeted-to-uri that was retargeted as opposed to the hi-targeted-to-uri to which it was retargeted.

- Privacy: An optional parameter for History-Info, reflected in the History-Info header field values by including the Privacy Header [RFC3323] escaped in the hi- targeted-to-uri or by adding the Privacy header to the request. The latter case indicates that the History-Info entries for the domain MUST be anonymized prior to forwarding, whereas the use of the Privacy header escaped in the hi-targeted-to-uri means that a specific hi-entry MUST be anonymized.

- Target (hi-target): An optional parameter for the History-Info header. This parameter is added if the URI to which the outgoing request is targeted contains a "hit" URI parameter Section 6.3.1. The hi-target is added for a hi-entry when it is first added in a History-Info header field, and only one value is permitted. Upon receipt of a request or response containing the History-Info header, a UA can determine the mechanism by which the target was determined. The following attributes are defined for this parameter derived from the values of the "hit" URI parameter:

  * "rc": The hi-target is set to "rc" if the "hit" URI parameter is set to a value of "rc".

  * "mp": The hi-target is set to "mp" if the "hit" URI parameter is set to a value of "mp".The "mp" attribute in the hi-target parameter includes the index parameter for the hi-targeted-to-uri that was retargeted, thus identifying the "mapped from" target.

- Extension (hi-extension): A parameter to allow for future optional extensions. As per [RFC3261], any implementation not understanding an extension should ignore it.

The following summarizes the syntax of the History-Info header, based upon the standard SIP syntax [RFC3261]:

```
History-Info = "History-Info" HCOLON hi-entry *(COMMA hi-entry)

hi-entry = hi-targeted-to-uri *(SEMI hi-param)

hi-targeted-to-uri = name-addr

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

hi-index = "index" EQUAL 1*DIGIT *("." 1*DIGIT)

hi-target = "rc" / mp-param

mp-param = "mp" EQUAL 1*DIGIT *("." 1*DIGIT)

hi-extension = generic-param
```

The following rules apply:

- There MUST be exactly one hi-index parameter per hi-entry.
- There MUST be no more than one hi-target parameter.
- There MAY be any number of hi-extension parameters.
- The ABNF definitions for "generic-param" and "name-addr" are from [RFC3261].

6.2. Examples

The following provides some examples of the History-Info header. Note that the backslash and CRLF between the fields in the examples below are for readability purposes only.

```
History-Info: <sip:UserA@ims.example.com>;index=1;foo=bar

History-Info: <sip:UserA@ims.example.com?Reason=SIP%3B\
              cause%3D302>;index=1.1,\
              <sip:UserB@example.com?Privacy=history&Reason=SIP%3B\
              cause%3D486>;index=1.2;mp=1.1,\
              <sip:45432@192.168.0.3>;index=1.3;rc
```

6.3. Procedures

The following sections define procedures for processing of the History-Info header and the "hit" SIP/SIPS URI parameter and Privacy header. These procedures may be applicable to processing entities such as Proxies, Redirect Servers or User Agents.

6.3.1. SIP/SIPS URI target parameter for History-Info Header

This specification defines a new SIP/SIPS URI Parameter indicating the mechanism by which a new target for a request is determined. This parameter is added to the SIP/SIPS URIs as they are added to the target set per the procedures of 16.5 [RFC3261]] or by a redirect server as it populates the Contact header for a 3xx response. This parameter is used to populate the hi-target parameter Section 6.3.5 when a Request-URI is first added in a hi-entry in the History-Info header field. This parameter MUST be removed from the URI when the hi-entry is constructed.

The following two values are defined for this URI parameter:

- "rc": The Request-URI is a contact that is bound to an AOR in an abstract location service. The AOR-to-contact binding has been placed into the location service by a SIP Registrar that received a SIP REGISTER request.

- "mp": The Request-URI is a URI that represents another user. This occurs when a request is to be statically or dynamically retargeted to another user.

The following defines the ABNF for the "hit" URI parameter:

```
hit-param      = "hit=" ("rc" / "mp")
```

6.3.2. Privacy in the History-Info Header

The privacy requirements for this document are described in Appendix A.2.

The History-Info header can inadvertently reveal information about the requestor as described in [RFC3323]. The Privacy header is used to ensure the privacy requirements are satisfied for both the History-Info entries received by the intermediary in the incoming request (per PRIV-req-2 in Appendix A.2) and the History-Info entries that are added by the intermediary as the request is retargeted per PRIV-req-1 in Appendix A.2. If the requestor has indicated a priv-value of "session" or "header" in the request, all History-Info entries MUST be anonymized when the request leaves a domain for which the intermediary is responsible.

Privacy can also be associated with a specific History-Info entry, and any entry that corresponds to that same user rather than all History-Info entries in a request. For example, if Alice sends a

request to Bob without any privacy, and Bob redirects to Carol with privacy setup for himself, Carol should receive a request where Alice's history information is present, but Bob's has been anonymized. This is accomplished by adding a new priv-value, history, to the Privacy header [RFC3323] indicating that a specific History-Info header entry can not be forwarded outside the domain.

In addition, the History-Info header can reveal general routing information which may be viewed by a specific intermediary or network. Thus, a proxy can use local policy to determine whether the History-Info header entries for it's whole domain are private or not when exiting the domain through retargeting (PRIV-req-3).

It is recognized that satisfying the privacy requirements can impact the functionality of this solution by overriding the request to generate the information. Thus, the following recommendations are made to ensure optimal functionality without compromising privacy:

If there is a Privacy header in the request with a priv-value of "session", "header", or "history", an hi-entry SHOULD be added if the request is being retargeted to a URI associated with a domain for which the processing entity is responsible. If there is no Privacy header, but the processing entity's local policies indicate that the hi-entry(s) cannot be forwarded beyond the domain for which this intermediary is responsible, then a Privacy header with a priv-value of "history" SHOULD be associated with each hi-entry added by the proxy as the request is forwarded within the domain.

If a request is being retargeted to a URI associated with a domain for which the processing identity is not responsible and there is a Privacy header in the request with a priv-value of "session", "header", or "history", the processing entity MUST anonymize hi-entry(s) as per [RFC3323] prior to forwarding, unless the processing entity knows a priori that it can rely on a downstream processing entity within its domain to apply the requested privacy or local policy allows the forwarding.

6.3.3. Reason in the History-Info Header

For retargets that are the result of an explicit SIP response, a Reason MUST be associated with the hi-targeted-to-uri. If the SIP response does not include a Reason header (see [RFC3326]), the SIP Response Code that triggered the retargeting MUST be included as the Reason associated with the hi-targeted-to-uri that has been retargeted. If the response contains a Reason header for a protocol that is not SIP (e.g., Q.850), it MUST be captured as an additional Reason associated with the hi-targeted-to-uri that has been

retargeted, along with the SIP Response Code. If the Reason header is a SIP reason, then it MUST be used as the Reason associated with the hi-targeted-to-uri rather than the SIP response code.

If a request has timed out (instead of being explicitly rejected), it SHOULD be treated as if a 487 "Request Terminated" error response code was received.

6.3.4. Indexing in the History-Info Header

In order to maintain ordering and accurately reflect the nesting and retargeting of the request, an index MUST be included along with the Targeted-to-URI being captured. Per the syntax in Section 6, the index consists of a dot-delimited series of digits (e.g., 1.1.2). Each dot reflects a hop or level of nesting; thus, the number of hops is determined by the total number of dots. Within each level, the integer reflects the number of peer entities to which the request has been routed. Thus, the indexing results in a logical tree representation for the history of the request. For each level of indexing, the index MUST start at 1. An increment of 1 MUST be used for advancing to a new branch. The first entry MUST be set to 1.

The basic rules for adding the index are summarized as follows:

1. Basic Forwarding: In the case of a request that is being forwarded, the index is determined by adding another sub-level of indexing since the depth/length of the branch is increasing. To accomplish this, the processing entity reads the value from the History-Info header in the received request, if available, and adds another level of indexing by appending the dot delimiter followed by an initial index for the new level of 1. For example, if the index in the last History-Info header field in the received request is 1.1, this proxy would initialize its index to 1.1.1 and forward the request.

2. Retargeting within a processing entity - 1st instance: For the first instance of retargeting within a processing entity, the calculation of the index follows that prescribed for basic forwarding.

3. Retargeting within a processing entity - subsequent instance: For each subsequent retargeting of a request by the same processing entity, another branch is added. With the index for each new branch calculated by incrementing the last/lowest digit at the current level, the index in the next request forwarded by this same processing entity, following the example above, would be 1.1.2.

4. Retargeting based upon a Response: In the case of retargeting due to a specific response (e.g., 302), the index would be calculated per rule 3. That is, the lowest/last digit of the index is incremented (i.e., a new branch is created), with the increment of 1. For example, if the index in the History-Info header of the received request was 1.2, then the index in the History-Info header field for the new hi-targeted- to-URI would be 1.3.

5. Forking requests: If the request forwarding is done in multiple forks (sequentially or in parallel), the index MUST be captured for each forked request per the rules above, with each new request having a unique index. Each index are sequentially assigned. For example, if the index in the last History-Info header field in the received request is 1.1, this processing entity would initialize its index to 1.1.1 for the first fork, 1.1.2 for the second, and so forth (see Figure 1 for an example). Note that for each individual fork, only the entry corresponding that that fork is included (e.g., the entry for fork 1.1.1 is not included in the request sent to fork 1.1.2, and vice-versa).

6. When a response is built and it represents the aggregate of multiple forks (e.g., multiple forks that fail), the processing entity builds the subsequent response using the aggregated information associated with each of those forks and including the header entries in the order indicated by the indexing. For example, if a procesing entity received failure responses for forks 1.1.1 and 1.1.2, it would forward both the 1.1.1 and 1.1.2 entries to 1.1. See Appendix B.1 for an example. Responses are processed as described in Section 16.7 of [RFC3261] with the aggregated History-Info entries processed similar to Step 7 "Aggregate Authentication Header Field Values".

6.3.5. Request Target in the History-Info Header

The hi-target attribute MUST be added to the History-Info header if the target to which the request is being forwarded contains a "hit" URI parameter as defined in Section 6.3.1.

If the value of the "hit" parameter is "mp", then the index of the entry corresponding to the original target (i.e., the "mapped-from" target) MUST be added as a parameter to "mp".

7. Application Considerations

History-Info provides a very flexible building block that can be used by intermediaries and UAs for a variety of services. The following summarizes the categories of information that applications may use:

1. Complete history information - e.g., for debug or other operational and management aspects, optimization of determining targets to avoid retargeting to the same URI, etc. This information is relevant to proxies, UACs and UASs.

2. Entry prior to last entry with hi-target of "rc" in the Request received by a UAS - i.e., the Request URI associated with the destination of the request was determined based on a Registered Contact.

3. Entry with the index that matches the value of the last entry with a "mp" hi-target parameter in the Request received by a UAS - i.e., the last Request URI that was mapped to reach the destination.

4. Entry prior to first entry with hi-target of "rc" in the Request received by a UAS - i.e., the first Registered Contact to which the request was targeted.

5. Entry with the index that matches the value of the first entry with a "mp" hi-target parameter - i.e., the original target of the request.

In many cases, applications are most interested in the information within a particular domain(s), thus only a subset of the information is required.

Some applications may use multiple types of information. For example, an Automatic Call Distribution (ACD)/Call center application that utilizes the entry prior to the first History-Info entry with an hi-target value of "mp", may also display other agents, reflected by other History-Info entries prior to entries with hi-target values of "rc", to whom the call was targeted prior to its arrival at the current agent. This could allow the agent the ability to decide how they might forward or reroute the call if necessary (avoiding agents that were not previously available for whatever reason, etc.).

Since support for History-Info header is optional, a service MUST define default behavior for requests and responses not containing History-Info headers. For example, an entity may receive only partial History-Info entries or entries which are not tagged appropriately with an hi-target parameter. This may not impact some applications (e.g., debug), however, it could require some applications to make some default assumptions in this case. For example, in an ACD scenario, the application could select the oldest hi-entry with the domain associated with the ACD system and display that as the original called party. Depending upon how and where the request may have been retargeted, the complete list of agents to whom

the call was targeted may not be available.

8. Security Considerations

The security requirements for this document are specified in Appendix A.1.

This document defines a header for SIP. The use of the Transport Layer Security (TLS) protocol [RFC5246] as a mechanism to ensure the overall confidentiality of the History-Info headers (SEC-req-4) is strongly RECOMMENDED. This results in History-Info having at least the same level of security as other headers in SIP that are inserted by intermediaries. With TLS, History-Info headers are no less, nor no more, secure than other SIP headers, which generally have even more impact on the subsequent processing of SIP sessions than the History-Info header.

With the level of security provided by TLS (SEC-req-3), the information in the History-Info header can thus be evaluated to determine if information has been removed by evaluating the indices for gaps (SEC-req-1, SEC-req-2). It would be up to the application to define whether it can make use of the information in the case of missing entries.

Note that while using the SIPS scheme (as per [RFC5630]) protects History-Info from tampering by arbitrary parties outside the SIP message path, all the intermediaries on the path are trusted implicitly. A malicious intermediary could arbitrarily delete, rewrite, or modify History-Info. This specification does not attempt to prevent or detect attacks by malicious intermediaries.

9. IANA Considerations

This document requires several IANA registrations detailed in the following sections.

This document updates [RFC4244] but uses the same SIP header field name and option tag. The IANA registry needs to update the references to [RFC4244] with [RFCXXXX].

9.1. Registration of New SIP History-Info Header

This document defines a SIP header field name: History-Info and an option tag: histinfo. The following changes have been made to http:///www.iana.org/assignments/sip-parameters The following row has been added to the header field section:.

The following row has been added to the header field section:

Header Name	Compact Form	Reference
History-Info	none	[RFCXXXX]

The following has been added to the Options Tags section:

Name	Description	Reference
histinfo	When used with the Supported header, this option tag indicates the UAC supports the History Information to be captured for requests and returned in subsequent responses. This tag is not used in a Proxy-Require or Require header field since support of History-Info is optional.	[RFCXXXX]

Note to RFC Editor: Please replace RFC XXXX with the RFC number of this specification.

9.2. Registration of "history" for SIP Privacy Header

This document defines a priv-value for the SIP Privacy header: history The following changes have been made to http://www.iana.org/assignments/sip-priv-values The following has been added to the registration for the SIP Privacy header:

Name	Description	Registrant	Reference
history	Privacy requested for History-Info header(s)	Mary Barnes mary.barnes@nortel.com	[RFCXXXX]

Note to RFC Editor: Please replace RFC XXXX with the RFC number of this specification.

9.3. Registration of "hit" SIP/SIPS URI Parameter

This specification adds a new value to the IANA registration in the "SIP/SIPS URI Parameters" registry, http:/www.iana.org/assignments/sip-parameters, as defined in [RFC3969].

Parameter Name	Predefined Values	Reference

hit	Yes ("rc"/"mp")	[RFCxxxx]

Note to RFC Editor: Please replace RFC XXXX with the RFC number of this specification.

10. Acknowledgements

Jonathan Rosenberg et al produced the document that provided additional use cases precipitating the requirement for the new "target" parameter in the History-Info header and the new SIP/SIPS URI parameter. Ian Elz provided feedback on the privacy aspects.

Mark Watson, Cullen Jennings and Jon Peterson provided significant input into the initial work that resulted in the development of of [RFC4244]. The editor would like to acknowledge the constructive feedback provided by Robert Sparks, Paul Kyzivat, Scott Orton, John Elwell, Nir Chen, Palash Jain, Brian Stucker, Norma Ng, Anthony Brown, Jayshree Bharatia, Jonathan Rosenberg, Eric Burger, Martin Dolly, Roland Jesske, Takuya Sawada, Sebastien Prouvost, and Sebastien Garcin in the development of [RFC4244].

The editor would like to acknowledge the significant input from Rohan Mahy on some of the normative aspects of the ABNF for [RFC4244], particularly around the need for and format of the index and around the security aspects.

11. Changes from RFC 4244

This RFC replaces [RFC4244].

Deployment experience with [RFC4244] over the years has shown a number of issues, warranting an update:

- In order to make [RFC4244] work in "real life", one needs to make "assumptions" on how History-Info is used. For example, many implementations filter out many entries, and only leave specific entries corresponding, for example, to first and last redirection. Since vendors uses different rules, it causes significant interoperability isssues.

- [RFC4244] is overly permissive and evasive about recording entries, causing interoperability issues.

- The examples in the call flows had errors, and confusing because they often assume "loose routing".

- [RFC4244] has lots of repetitive and unclear text due to the combination of requirements with solution.

- [RFC4244] gratuitly mandates the use of TLS on every hop. No existing implementation enforces this rule, and instead, the use of TLS or not is a general SIP issue, not an [RFC4244] issue per se.

- [RFC4244] does not include clear procedures on how to deliver current target URI information to the UAS when the Request-URI is replaced with a contact.

- [RFC4244] does not allow for marking History-Info entries for easy processing by User Agents.

The following summarizes the functional changes between this specification and [RFC4244]:

1. Added a parameter, "target", to the History-Info header and defined a SIP/SIPS URI parameter indicating targets that are determined by a registered contact or based on mapping the incoming Request-URI to another user to facilitate processing at the UAS. The URI parameter is added to the target URIs as the target set is determined (per section 16.5 of [RFC3261]) and removed when the "target" parameter is populated, using the URI parameter value, in the History-Info header (per section 16.6 of [RFC3261]).

2. Rather than recommending that entries be removed in the case of certain values of the privacy header, recommend that the entries are anonymized.

3. Updated the security section to be equivalent to the security recommendations for other SIP headers inserted by intermediaries.

The first 2 changes are intended to facilitate application usage of the History-Info header and eliminate the need to make assumptions based upon the order of the entries and ensure that the most complete set of information is available to the applications.

In addition, editorial changes were done to both condense and clarify the text, moving the requirements to an appendix. The examples were simplified and updated to reflect the protocol changes. Several of the call flows in the appendix were removed and put into a separate document that includes additional use cases that require the new

parameters.

11.1. Backwards compatibility

This specification is backwards compatible since [RFC4244] allows for the addition of new optional parameters. This specification adds an optional parameter "target" to the History-Info header. Entities that have not implemented this specification should ignore this parameter, however, per [RFC4244] an entity MUST NOT remove this parameter from an hi-entry. This specification defines a SIP/SIPS URI parameter, "hit". An entity that does not understand the "hit" URI parameter SHOULD ignore the parameter and MAY remove the parameter from the URI as it is used as the target for a request.

12. Changes since last Version

NOTE TO THE RFC-Editor: Please remove this section prior to publication as an RFC.

Changes from 00 to 01:

1. Moved examples (except first) in appendix to a new (informational) document.

2. Updated UAS and UAC sections to clarify and expand on the handling of the History-Info header.

3. Updated the Application considerations section:

   * Included more detail with regards to how applications can make use of the information, in particular based on the new tags.

   * Removed privacy consideration (2nd bullet) since privacy is now accomplished by anonymizing rather than removal of entries.

Changes from (individual) barnes-sipcore-4244bis-03 to (WG) ietf-sipcore-4244bis-00:

1. Added a new SIP/SIPS URI parameter to tag the URIs as they are added to the target list and those returned in the contact header in a 3xx response.

2. Updated description of "target" parameter to use the new URI parameter value in setting the value for the parameter.

3. Clarified privacy.

4. Changed handling at redirect server to include the use of the new URI parameter and to remove the functionality of adding the History-Info entries (basically reverting to core 4244 processing).

5. Additional text to clarify that a service such as voicemail can be done in multiple ways.

6. Editorial changes including removal of some vestiges of tagging all entries (including the "aor" tag).

Changes from barnes-sipcore-4244bis-02 to 03:

1. Fixed problem with indices in example in voicemail example.

2. Removed oc and rt from the Hi-target parameter.

3. Removed aor tag

4. Added index parameter to "mp"

5. Added use-cases and call-flows from target-uri into appendix.

Changes from barnes-sipcore-4244bis-01 to 02:

1. Added hi-aor parameter that gets marked on the "incoming" hi-entry.

2. Hi-target parameter defined to be either rc, oc, mp, rt, and now gets included when adding an entry.

3. Added section on backwards compatibility, as well as added the recognition and handling of requests that do not support this specification in the appropriate sections.

4. Updated redirect server/3xx handling to support the new parameters - i.e., the redirecting entity must add the new entry since the proxy does not have access to the information as to how the Contact was determined.

5. Added section on normative differences between this document and RFC 4244.

6. Restructuring of document to be more in line with current IETF practices.

7. Moved Requirements section into an Appendix.

8. Fixed ABNF to remove unintended ordering requirement on hi-index that was introduced in attempting to illustrate it was a mandatory parameter.

Changes from barnes-sipcore-4244bis-00 to 01 :

1. Clarified "retarget" definition.

2. Removed privacy discussion from optionality section - just refer to privacy section.

3. Removed extraneous text from target-parameter (leftover from sip-4244bis). Changed the terminology from the "reason" to the "mechanism" to avoid ambiguity with parameter.

4. Various changes to clarify some of the text around privacy.

5. Reverted proxy response handling text to previous form - just changing the privacy aspects to anonymize, rather than remove.

6. Other editorial changes to condense and simplify.

7. Moved Privacy examples to Appendix.

8. Added forking to Basic call example.

Changes from barnes-sip-4244bis-00 to barnes-sipcore-4244bis-00:

1. Added tags for each type of retargeting including proxy hops, etc. - i.e., a tag is defined for each specific mechanism by which the new Request-URI is determined. Note, this is extremely helpful in terms of backwards compatibility.

2. Fixed all the examples. Made sure loose routing was used in all of them.

3. Removed example where a proxy using strict routing is using History-Info for avoiding trying same route twice.

4. Remove redundant Redirect Server example.

5. Index are now mandated to start at "1" instead of recommended.

6. Clarified 3xx behavior as the entity sending the 3XX response MUST add the hi-target attribute to the previous hi-entry to ensure that it is appropriately tagged (i.e., it's the only one

that knows how the contact in the 3xx was determined.)

7. Removed lots of ambiguity by making many "MAYs" into "SHOULDs" and "some "SHOULDs" into "MUSTs".

8. Privacy is now recommended to be done by anonymizing entries as per RFC 3323 instead of by removing or omitting hi-entry(s).

9. Requirement for TLS is now same level as per RFC 3261.

10. Clarified behavior for "Privacy" (i.e., that Privacy is for Hi-entries, not headers).

11. Removed "OPTIONALITY" as specific requirements, since it's rather superflous.

12. Other editorial changes to remove redundant text/sections.

Changes from RFC4244 to barnes-sip-4244bis-00:

1. Clarified that HI captures both retargeting as well as cases of just forwarding a request.

2. Added descriptions of the usage of the terms "retarget", "forward" and "redirect" to the terminology section.

3. Added additional examples for the functionality provided by HI for core SIP.

4. Added hi-target parameter values to HI header to ABNF and protocol description, as well as defining proxy, UAC and UAS behavior for the parameter.

5. Simplified example call flow in section 4.5. Moved previous call flow to appendix.

6. Fixed ABNF per RFC4244 errata "dot" -> "." and added new parameter.

13. References

13.1. Normative References

[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.

[RFC3326] Schulzrinne, H., Oran, D., and G. Camarillo, "The Reason Header Field for the Session Initiation Protocol (SIP)", RFC 3326, December 2002.

[RFC3323] Peterson, J., "A Privacy Mechanism for the Session Initiation Protocol (SIP)", RFC 3323, November 2002.

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

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC4244] Barnes, M., "An Extension to the Session Initiation Protocol (SIP) for Request History Information", RFC 4244, November 2005.

13.2. Informative References

[RFC5627] Rosenberg, J., "Obtaining and Using Globally Routable User Agent URIs (GRUUs) in the Session Initiation Protocol (SIP)", RFC 5627, October 2009.

[RFC5630] Audet, F., "The Use of the SIPS URI Scheme in the Session Initiation Protocol (SIP)", RFC 5630, October 2009.

[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.

[RFC3969] Camarillo, G., "The Internet Assigned Number Authority (IANA) Uniform Resource Identifier (URI) Parameter Registry for the Session Initiation Protocol (SIP)", BCP 99, RFC 3969, December 2004.

Appendix A. Request History Requirements

The following list constitutes a set of requirements for a "Request History" capability.

1. CAPABILITY-req: The "Request History" capability provides a capability to inform proxies and UAs involved in processing a request about the history/progress of that request. Although this is inherently provided when the retarget is in response to a SIP redirect, it is deemed useful for non-redirect retargeting

scenarios, as well.

2. GENERATION-req: "Request History" information is generated when the request is retargeted.

   A. In some scenarios, it might be possible for more than one instance of retargeting to occur within the same Proxy. A proxy should also generate Request History information for the 'internal retargeting'.

   B. An entity (UA or proxy) retargeting in response to a redirect or REFER should include any Request History information from the redirect/REFER in the new request.

3. ISSUER-req: "Request History" information can be generated by a UA or proxy. It can be passed in both requests and responses.

4. CONTENT-req: The "Request History" information for each occurrence of retargeting shall include the following:

   A. The new URI or address to which the request is in the process of being retargeted,

   B. The URI or address from which the request was retargeted, and wether the retarget URI was an AOR

   C. The mechanism by which the new URI or address was determined,

   D. The reason for the Request-URI or address modification,

   E. Chronological ordering of the Request History information.

5. REQUEST-VALIDITY-req: Request History is applicable to requests not sent within an established dialog (e.g., INVITE, REGISTER, MESSAGE, and OPTIONS).

6. BACKWARDS-req: Request History information may be passed from the generating entity backwards towards the UAC. This is needed to enable services that inform the calling party about the dialog establishment attempts.

7. FORWARDS-req: Request History information may also be included by the generating entity in the request, if it is forwarded onwards.

A.1. Security Requirements

The Request History information is being inserted by a network element retargeting a Request, resulting in a slightly different problem than the basic SIP header problem, thus requiring specific consideration. It is recognized that these security requirements can be generalized to a basic requirement of being able to secure information that is inserted by proxies.

The potential security problems include the following:

1. A rogue application could insert a bogus Request History entry either by adding an additional entry as a result of retargeting or entering invalid information.

2. A rogue application could re-arrange the Request History information to change the nature of the end application or to mislead the receiver of the information.

3. A rogue application could delete some or all of the Request History information.

Thus, a security solution for "Request History" must meet the following requirements:

1. SEC-req-1: The entity receiving the Request History must be able to determine whether any of the previously added Request History content has been altered.

2. SEC-req-2: The ordering of the Request History information must be preserved at each instance of retargeting.

3. SEC-req-3: The entity receiving the information conveyed by the Request History must be able to authenticate the entity providing the request.

4. SEC-req-4: To ensure the confidentiality of the Request History information, only entities that process the request should have visibility to the information.

It should be noted that these security requirements apply to any entity making use of the Request History information.

A.2. Privacy Requirements

Since the Request-URI that is captured could inadvertently reveal information about the originator, there are general privacy requirements that MUST be met:

1. PRIV-req-1: The entity retargeting the Request must ensure that it maintains the network-provided privacy (as described in [RFC3323]) associated with the Request as it is retargeted.

2. PRIV-req-2: The entity receiving the Request History must maintain the privacy associated with the information. In addition, local policy at a proxy may identify privacy requirements associated with the Request-URI being captured in the Request History information.

3. PRIV-req-3: Request History information subject to privacy shall not be included in ougoing messages unless it is protected as described in [RFC3323].

Appendix B. Example call flows

The scenarios in this section provide sample use cases for the History-Info header for informational purposes only. They are not intended to be normative. A basic forking use case is included, along with two use cases illustrating the use of the privacy.

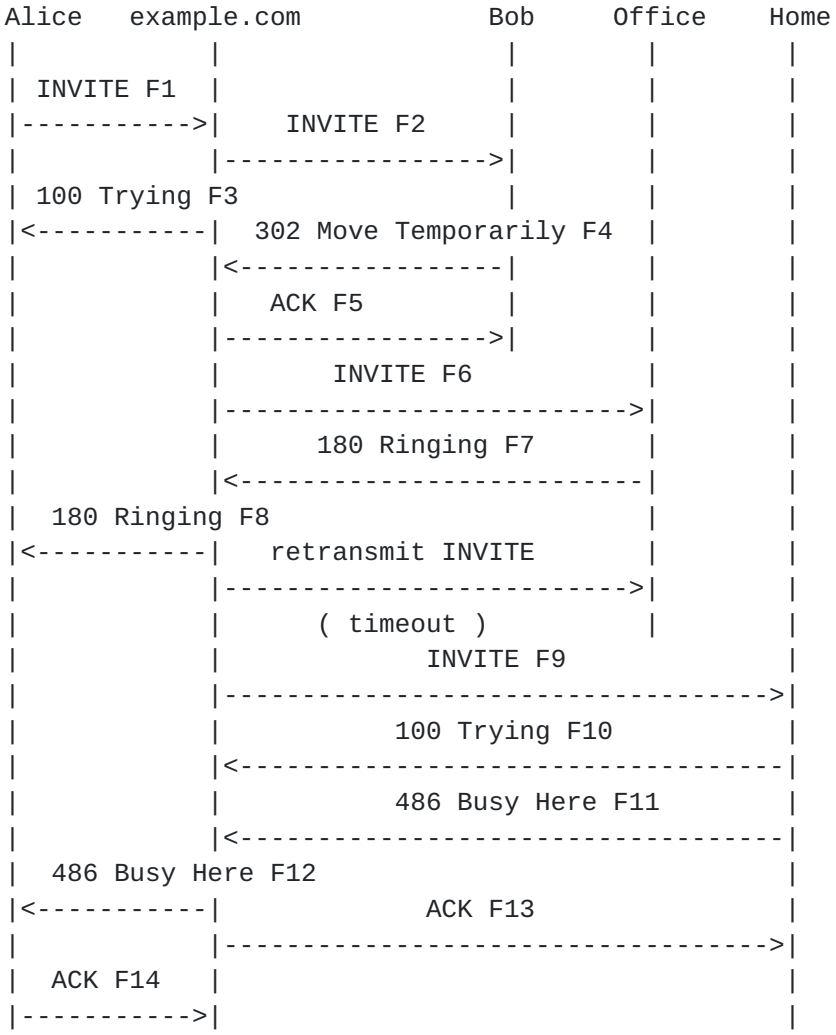
B.1. Sequentially Forking (History-Info in Response)

This scenario highlights an example where the History-Info in the response is useful to an application or user that originated the request.

Alice sends a call to Bob via sip:example.com. The proxy sip: example.com sequentially tries Bob on a SIP UA that has bound a contact with the sip:bob@example.com AOR, and then several alternate addresses (Office and Home) unsuccessfully before sending a response to Alice. The hi-entry containing the initial contact is the entry just prior to the firt entry tagged with an hi-target value of "rc". In this example, the Office and Home are not the same AOR as sip:bob@example.com, but rather different AORs that have been configured as alternate addresses for Bob in the proxy. In other words, Office and Bob are not bound through SIP Registration with Bob's AOR. This type of arrangement is common for example when a "routing" rule to a PSTN number is manually configured in a Proxy. These hi-entries are identified by the index contained in the hi-target "mp" parameter in the hi-entries.

This scenario illustrates that by providing the History-Info to Alice, the end-user or an application at Alice could make a decision on how best to attempt finding Bob without sending multiple requests to the same destination. Upon receipt of the response containing the History-Info entries, the Request URIs for the History-Info entries

tagged with "mp" are extracted. Those Request-URIs can be compared to other URIs (if any) that might be attempted in order to establish the session with Bob. Thus, avoiding another INVITE to Bob's home phone. Without this mechanism, Alice might well attempt to reach Bob at his office phone, which would then retarget the request to Bob's home phone. When that attempt failed, then Alice might attempt to reach Bob directly at his home phone, unknowingly for a third time.



Message Details

F1 INVITE alice -> example.com

```
INVITE sip:alice@example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Supported: histinfo
Call-Id: 12345600@example.com
CSeq: 1 INVITE
Contact: Alice <sip:alice@192.0.2.3>
Content-Type: application/sdp
Content-Length: <appropriate value>
<!-- SDP Not Shown -->
```

F2 INVITE example.com -> Bob

```
INVITE sip:bob@192.0.2.4 SIP/2.0
Via: SIP/2.0/TCP proxy.example.com:5060
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Supported: histinfo
Call-Id: 12345600@example.com
CSeq: 1 INVITE
Record-Route: <sip:proxy.example.com;lr>
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4>;index=1.1;rc
Contact: Alice <sip:alice@192.0.2.3>
Content-Type: application/sdp
Content-Length: <appropriate value>
<!-- SDP Not Shown -->
```

F3 100 Trying example.com -> alice

```
SIP/2.0 100 Trying
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
CSeq: 1 INVITE
Content-Length: 0
```

F4 302 Moved Temporarily Bob -> example.com

```
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/TCP proxy.example.com:5060
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>;tag=3
Call-Id: 12345600@example.com
CSeq: 1 INVITE
Record-Route: <sip:proxy.example.com;lr>
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4>;index=1.1;rc
Contact: <sip:office@example.com;hit=mp>
Content-Length: 0
```

F5 ACK 192.0.2.4 -> Bob

```
ACK sip:home@example.com SIP/2.0
Via: SIP/2.0/TCP proxy.example.com:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
CSeq: 1 ACK
Content-Length: 0
```

F6 INVITE example.com -> office

```
INVITE sip:office@192.0.2.3.com SIP/2.0
Via: SIP/2.0/TCP proxy.example.com:5060;branch=2
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Supported: histinfo
Call-Id: 12345600@example.com
Record-Route: <sip:proxy.example.com;lr>
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
              index=1.1;rc
History-Info: <sip:office@example.com>;index=1.2;mp=1
History-Info: <sip:office@192.0.2.5>;index=1.2.1
CSeq: 1 INVITE
Contact: Alice <sip:alice@192.0.2.3>
Content-Type: application/sdp
Content-Length: <appropriate value>
```

```
   <!-- SDP Not Shown -->
```

F7 180 Ringing office -> example.com

```
   SIP/2.0 180 Ringing
   Via: SIP/2.0/TCP proxy.example.com:5060;branch=2
   Via: SIP/2.0/TCP 192.0.2.3:5060
   From: Alice <sip:alice@example.com>
   To: Bob <sip:bob@example.com>;tag=5
   Supported: histinfo
   Call-ID: 12345600@example.com
   Record-Route: <sip:proxy.example.com;lr>
   History-Info: <sip:bob@example.com>;index=1
   History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
                 index=1.1;rc
   History-Info: <sip:office@example.com>;index=1.2;mp=1
   History-Info: <sip:office@192.0.2.5>;index=1.2.1
   CSeq: 1 INVITE
   Content-Length: 0
```

F8 180 Ringing example.com -> alice

```
   SIP/2.0 180 Ringing
   Via: SIP/2.0/TCP example.com:5060
   From: Alice <sip:alice@example.com>
   To: Bob <sip:bob@example.com>
   Supported: histinfo
   Call-Id: 12345600@example.com
   History-Info: <sip:bob@example.com>;index=1
   History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
                 index=1.1;rc
   History-Info: <sip:office@example.com>;index=1.2;mp=1
   History-Info: <sip:office@192.0.2.5>;index=1.2.1
   CSeq: 1 INVITE
   Content-Length: 0
```

F9 INVITE example.com -> home

```
INVITE sip:home@192.0.2.6 SIP/2.0
Via: SIP/2.0/TCP proxy.example.com:5060;branch=3
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Supported: histinfo
Call-Id: 12345600@example.com
Record-Route: <sip:proxy.example.com;lr>
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
              index=1.1;rc
History-Info: <sip:office@example.com>;index=1.2;mp=1
History-Info: <sip:office@192.0.2.5?Reason=SIP;cause=480>;\
              index=1.2.1>;index=1.2.1
History-Info: <sip:home@example.com>;index=1.3;mp=1.2
History-Info: <sip:home@192.0.2.6>;index=1.3.1
CSeq: 1 INVITE
Contact: Alice <sip:alice@192.0.2.3>
Content-Type: application/sdp
Content-Length: <appropriate value>
<!-- SDP Not Shown -->
```

F10 100 Trying home -> example.com

```
SIP/2.0 100 Trying
Via: SIP/2.0/TCP proxy.example.com:5060;branch=3
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
CSeq: 1 INVITE
Content-Length: 0
```

F11 486 Busy Here home -> example.com

```
SIP/2.0  486 Busy Here
Via: SIP/2.0/TCP proxy.example.com:5060;branch=3
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
Record-Route: <sip:proxy.example.com;lr>
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
              index=1.1;rc
History-Info: <sip:office@example.com>;index=1.2;mp=1
History-Info: <sip:office@192.0.2.5?Reason=SIP;cause=480>;\
              index=1.2.1>;index=1.2.1
History-Info: <sip:home@example.com>;index=1.3;mp=1.2
History-Info: <sip:home@192.0.2.6>;index=1.3.1
CSeq: 1 INVITE
Content-Length: 0
```

F12 486 Busy Here example.com -> alice

```
SIP/2.0  486 Busy Here
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
History-Info: <sip:bob@example.com>;index=1
History-Info: <sip:bob@192.0.2.4?Reason=SIP;cause=302>;\
              index=1.1;rc
History-Info: <sip:office@example.com>;index=1.2;mp=1
History-Info: <sip:office@192.0.2.5?Reason=SIP;cause=480>;\
              index=1.2.1>;index=1.2.1
History-Info: <sip:home@example.com>;index=1.3;mp=1.2
History-Info: <sip:home@192.0.2.6>;index=1.3.1
CSeq: 1 INVITE
Content-Length: 0
```

F13 ACK example.com -> home

```
ACK sip:home@example.com SIP/2.0
Via: SIP/2.0/TCP proxy.example.com:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
CSeq: 1 ACK
Content-Length: 0
```

F14 ACK alice -> example.com

```
ACK sip:bob@example.com SIP/2.0
Via: SIP/2.0/TCP 192.0.2.3:5060
From: Alice <sip:alice@example.com>
To: Bob <sip:bob@example.com>
Call-Id: 12345600@example.com
Route: <sip:proxy.example.com;lr>
CSeq: 1 ACK
Content-Length: 0
```

B.2. History-Info with Privacy Header

This example provides a basic call scenario without forking, with sip:biloxi.example.com adding the Privacy header indicating that the History-Info header information is anonymized outside the biloxi.example.com domain.

Figure 2: Example with Privacy Header

B.3. Privacy Header for a Specific History-Info Entry

This example provides a basic call scenario similar to Appendix B.2, however, due to local policy at sip:biloxi.example.com, only the final hi-entry in the History-Info, which is Bob's local URI, contains a priv-value of "history", thus providing Alice with some

information about the history of the request, but anonymizing Bob's local URI.

Figure 3: Example with Privacy Header for Specific URI

Authors' Addresses

Mary Barnes
Polycom
TX
US

Email: mary.ietf.barnes@gmail.com

Francois Audet
Skype

Email: francois.audet@skype.net

Shida Schubert
NTT

Email: shida@agnada.com

Hans Erik van Elburg
Detecon International Gmbh
Oberkasseler str. 2
Bonn,
Germany

Email: ietf.hanserik@gmail.com

Christer Holmberg
Ericsson
Hirsalantie 11, Jorvas
Finland

Email: christer.holmberg@ericsson.com