Network Working Group
Internet-Draft
Obsoletes: 4244 (if approved)
Intended status: Standards Track
Expires: April 4, 2014

M. Barnes
Polycom
F. Audet
Skype
S. Schubert
NTT
J. van Elburg
Detecon International Gmbh
C. Holmberg
Ericsson
Oct 2013

An Extension to the Session Initiation Protocol (SIP) for Request History Information
draft-ietf-sipcore-rfc4244bis-12.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 SIP request arrives at a specific application or user. This document defines an optional SIP header field, History-Info, for capturing the history information in requests. The document also defines SIP header field parameters for the History-Info and Contact header fields to tag the method by which the target of a request is determined. In addition, this specification defines a value for the Privacy header field that directs the anonymization of values in the History-Info header field. This document obsoletes RFC 4244.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on April 4, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

Table of Contents

1. Introduction 4
2. Conventions and Terminology 4
3. Background 5
4. Overview 6
5. History-Info Header Field Protocol Structure 7
5.1. History-Info Header Field Example Scenario 10
6. User Agent Handling of the History-Info Header Field 13
6.1. User Agent Client (UAC) Behavior 13
6.2. User Agent Server (UAS) Behavior 13
6.3. Back-2-Back User Agent (B2BUA) Behavior 13
7. Proxy/Intermediary Handling of History-Info Header Fields . . 14
8. Redirect Server Handling of History-Info Header Fields 14
9. Handling of History-Info Header Fields in Requests and Responses 15
9.1. Receiving a Request 15
9.2. Sending a Request with History-Info 15
9.3. Receiving a Response with History-Info or Request Timeouts 16
9.4. Sending History-Info in Responses 17
10. Processing the History-Info Header Field 17
10.1. Privacy in the History-Info Header Field 17
10.1.1. Indicating Privacy 17
10.1.2. Applying Privacy 18
10.2. Reason in the History-Info Header Field 19
10.3. Indexing in the History-Info Header Field 20
10.4. Mechanism for Target Determination in the History-Info Header Field 21
11. Application Considerations 23
12. Application Specific Usage 25
12.1. PBX Voicemail 25
12.2. Consumer Voicemail 25
13. Security Considerations 26
14. IANA Considerations 26
14.1. Registration of New SIP History-Info Header Field 27
14.2. Registration of "history" for SIP Privacy Header Field . . 27
14.3. Registration of Header Field Parameters 28
15. Acknowledgements 28
16. Changes from RFC 4244 29
16.1. Backwards compatibility 30
17. References 32
17.1. Normative References 32
17.2. Informative References 32
Appendix A. Request History Requirements 33
A.1. Security Requirements 34
A.2. Privacy Requirements 35
Authors' Addresses 35

1. Introduction

Many services that SIP is anticipated to support require the ability to determine why and how a SIP request 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 retarget capabilities that enable SIP requests to be routed to chosen applications, there is a need for a standard mechanism within SIP for communicating the retargeting history of the requests. This "request history" information allows the receiving application to obtain information about how and why the SIP request arrived at the application/user.

This document defines a SIP header field, 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. SIP header field parameters are defined for the History-Info and Contact header fields to tag the method by which the target of a request is determined. This specification also defines a value, "history", for the Privacy header field. In addition a SIP option tag, "histinfo", is defined.

The History-Info header field provides a building block for development of SIP based applications and services. The requirements for the solution described in this specification are included in Appendix A. Example scenarios using the History-Info header field are available in [I-D.ietf-sipcore-rfc4244bis-callflows].

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 specification to refer to the process of a SIP entity changing the request-URI [RFC3261, section 7.1] in a request based on the rules for determining request targets as described in Section 16.5 of [RFC3261] and of the subsequent forwarding of that request as described in step 2 in section 16.6 of [RFC3261]. This includes changing the Request-URI due to a location service lookup and redirect processing. This also includes internal (to a Proxy/SIP intermediary) changes of the URI prior to forwarding of the request.

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

The terms "target user" is used in this specification as the human user associated with particular AoR or AoRs (in case the human user has multiple alias).

The references to "domain for which the SIP entity/Proxy/Intermediary is responsible" are consistent with and intended to convey the same context as the usage of that terminology in [RFC3261]. The applicability of History-Info to architectures or models outside the context of [RFC3261] is outside the scope of this specification.

3. Background

SIP implicitly provides retargeting capabilities that enable SIP requests 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 request is retargeted to process the request 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 for other protocols provide the ability for elements involved with the request to obtain additional information relating to how and why the request 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 relaying whereby the recipient obtains a detailed "trace of the path" of the message from originator to receiver, including the time of each relay.

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 registrar or a "home proxy" (a proxy serving as the terminal point for routing an address-of-record) 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].

4. Overview

The fundamental functionality provided by the request history information is the ability to inform proxies and user agents (UAs) involved in processing a request about the history or progress of that request. The solution is to capture the Request-URIs as a request is retargeted, in a SIP header field: History-Info. 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 field is added to a Request when a new request is created by a user agent client (UAC) or forwarded by a Proxy, or when the target of a request is changed. It is possible for the target of a request to be changed by the same proxy/SIP intermediary multiple times (referred to as 'internal retargeting'). A SIP entity changing the target of a request in response to a redirect also propagates any History-Info header field from the initial request in the new request. The ABNF and detailed description of the History-Info header field parameters along with examples, is provided in Section 5. Section 6, Section 7 and Section 8 provide the detailed handling of the History-Info header field by SIP User Agents, Proxies and Redirect Servers respectively.

This specification also defines three new SIP header field parameters, "rc", "mp" and "np", for the History-Info and Contact header fields, to tag the method by which the target of a request is determined. Further detail on the use of these header field parameters is provided in Section 5.

This specification also defines a priv-value for the Privacy header, "history", that requires anonymization of all the History-Info header field entries in a Request or to a specific History-Info header field hi-entry as described above. Further detail is provided in Section 10.1.

In addition a SIP option tag, "histinfo", is defined. The use of this option tag is described in Section 6.1.

5. History-Info Header Field Protocol Structure

The History-Info header field defined in this specification defines the usage in out-of-dialog requests or initial requests for a dialog (e.g., INVITE, REGISTER, MESSAGE, REFER and OPTIONS, PUBLISH and SUBSCRIBE, etc.) and any non-100 provisional or final responses to these requests.

The following provides details for the information that is captured in the History-Info header field entries for each target used for forwarding a request:

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

- hi-index: A mandatory parameter for History-Info reflecting the chronological order of the information, indexed to reflect the forking and retargeting of requests. The format for this parameter is a sequence of non-negative integers, 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 leaf. By adding the new entries in chronological order (i.e., following existing entries per the details in Section 10.3), including the index and sending the messages using a secure transport, the ordering of the History-Info header fields in the request is assured. 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.

- hi-target-param: An optional parameter reflecting the mechanism by which the Request URI captured in the hi-targeted-to-uri in the History-Info header field value (hi-entry) was determined. This parameter is either an "rc", "mp" or "np" header field parameter, which is interpreted as follows:

"rc": The hi-targeted-to-URI represents a change in Request-URI while the target user remains the same. This occurs for example when user has multiple AoRs as an alias. The "rc" header field parameter contains the value of the hi-index in the hi-entry with an hi-targeted-to-uri that reflects the Request-URI that was retargeted

"mp": The hi-targeted-to-URI represents a user other than the target user associated with the Request-URI in the incoming request that was retargeted. This occurs when a request is statically or dynamically retargeted to another user represented by an AoR unassociated with the AoR of the original target user. The "mp" header field parameter contains the value of the hi-index in the hi-entry with an hi-targeted-to-uri that reflects the Request-URI that was retargeted, thus identifying the "mapped from" target.

"np": The hi-targeted-to-URI represents that there was no change in Request-URI. This would apply for example when a proxy merely forwards a request to a next hop proxy and loose routing is used. The "np" header field parameter contains the value of the hi-index in the hi-entry with an hi-targeted-to-uri that reflects the Request-URI that was copied unchanged into the request represented by this hi-entry. That value will usually be the hi-index of the parent hi-entry of this hi-entry.

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

The ABNF syntax [RFC5234] for the History-Info header field and header field parameters is as follows:

```
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-param / hi-extension

hi-index = "index" EQUAL index-val

index-val =  number *("." number)

number =  [ %31-39 *DIGIT ] DIGIT

hi-target-param = rc-param / mp-param / np-param

rc-param = "rc" EQUAL index-val

mp-param = "mp" EQUAL index-val

np-param = "np" EQUAL index-val

hi-extension = generic-param
```

The ABNF definitions for "generic-param", "name-addr", "HCOLON", "COMMA", "SEMI" and "EQUAL" are from [RFC3261].

This document also extends the "contact-params" for the Contact header field as defined in [RFC3261] with the "rc", "mp" and "np" header field parameters defined above.

In addition to the parameters defined by the ABNF, an hi-entry may also include a Reason header field and/or a Privacy header field, which are both included in the "headers" component of the hi-targeted-to-uri as described below:

- Reason: An optional parameter for History-Info, reflected in the History-Info header field by including the Reason header field [RFC3326] included in the hi-targeted-to-uri. A reason is included in the hi-targeted-to-uri of an hi-entry to reflect information received in a response to the request sent to that URI.

- Privacy: An optional parameter for History-Info, reflected in the History-Info header field values by including the Privacy Header [RFC3323] with a priv-value of "history", as defined in this document, included in the hi- targeted-to-uri or by adding the

Privacy header field with a priv-value of "history" to the request. The latter case indicates that the History-Info entries for all History-Info entries whose hi-targeted-to-uri has the same domain as the domain for which the SIP entity processing the message is responsible MUST be anonymized prior to forwarding, whereas the use of the Privacy header field included in the hi-targeted-to-uri means that a specific hi-entry MUST be anonymized.

Note that since both the Reason and Privacy parameters are included in the hi-targeted-to-uri, these fields will not be available in the case that the hi-targeted-to-uri is a Tel-URI [RFC3966].

The following provides examples of the format for the History-Info header field. Note that the backslash, CRLF and whitespace between the lines in the examples below are inserted for readability purposes only. Note, however, that History-Info can be broken into multiple lines due to the SWS (sep whitespace) that is part of HCOLON, COMMA and SEMI, and there can be multiple History-Info header fields due to the rule of section 7.3 [RFC3261]. Additional detailed examples are available in [I-D.ietf-sipcore-rfc4244bis-callflows].

```
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=1.2
```

5.1. History-Info Header Field Example Scenario

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 proxy in her home domain (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 original target information or the initial request-URI, including any parameters in the request URI. Bob can recover that information by locating the last hi-entry with

an "rc" header field parameter. This "rc" header field parameter contains the index of the hi-entry containing the lost target information - i.e., the sip:bob@biloxi.example.com hi-entry with index=1.1. Note that in the 200 response to Alice, an hi-entry is not included for the fork to sip:bob@192.0.2.7 (index 1.1.1) since biloxi.example.com had not received a response from that fork at the time it sent the 200 OK that ultimately reached Alice.

Additional detailed examples are available in [I-D.ietf-sipcore-rfc4244bis-callflows].

Note: This example uses loose routing procedures.

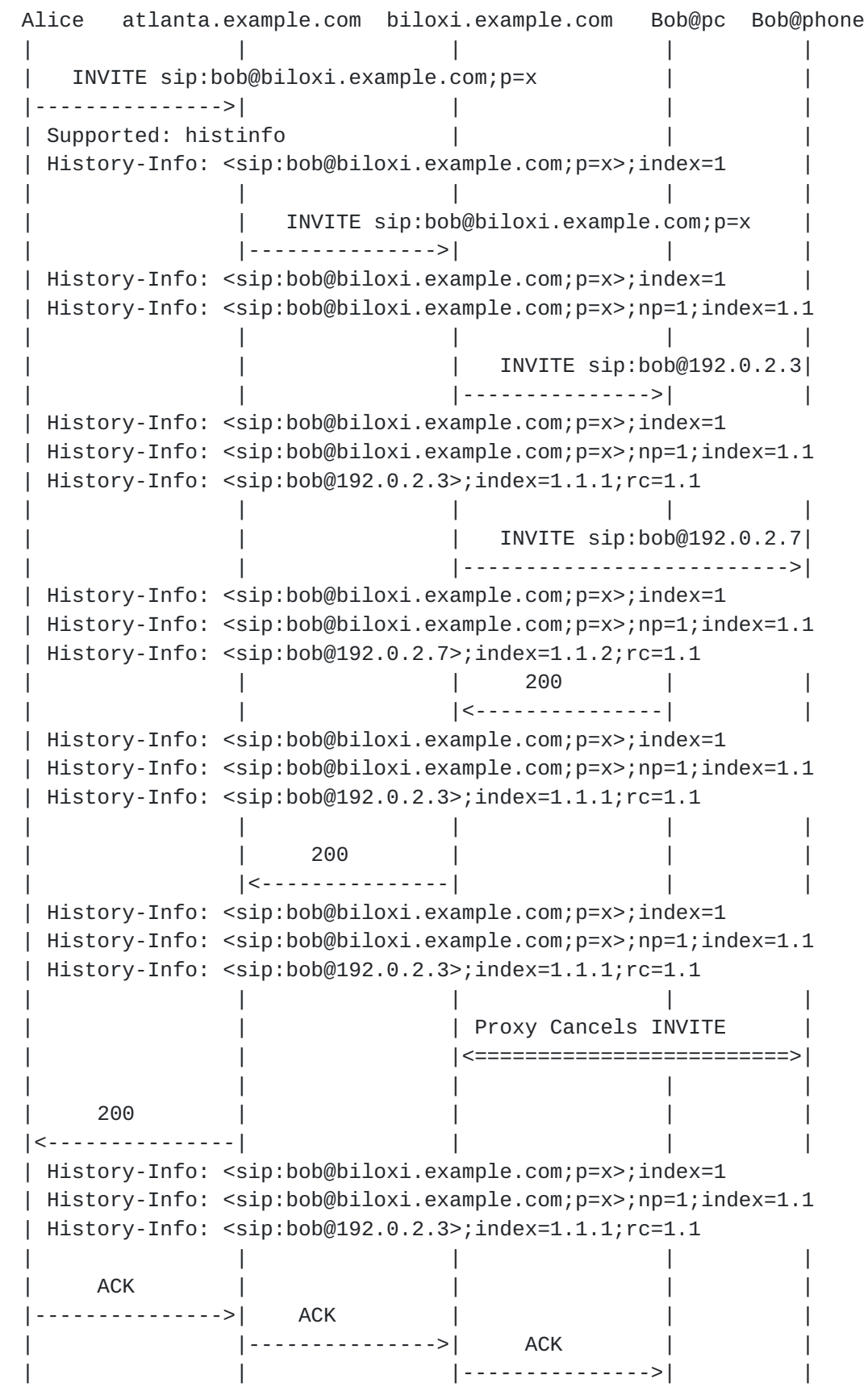

Alice
atlanta.example.com
biloxi.example.com
Bob@pc
Bob@phone
INVITE sip:bob@biloxi.example.com;p=x
Supported: histinfo
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
INVITE sip:bob@biloxi.example.com;p=x
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
INVITE sip:bob@192.0.2.3
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
History-Info: <sip:bob@192.0.2.3>;index=1.1.1;rc=1.1
INVITE sip:bob@192.0.2.7
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
History-Info: <sip:bob@192.0.2.7>;index=1.1.2;rc=1.1
200
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
History-Info: <sip:bob@192.0.2.3>;index=1.1.1;rc=1.1
200
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
History-Info: <sip:bob@192.0.2.3>;index=1.1.1;rc=1.1
Proxy Cancels INVITE
200
History-Info: <sip:bob@biloxi.example.com;p=x>;index=1
History-Info: <sip:bob@biloxi.example.com;p=x>;np=1;index=1.1
History-Info: <sip:bob@192.0.2.3>;index=1.1.1;rc=1.1
ACK
ACK
ACK

Figure 1: Basic Call

6. User Agent Handling of the History-Info Header Field

This section describes the processing specific to UAs(UACs, UASs and B2BUAs) for the History-Info header."

6.1. User Agent Client (UAC) Behavior

The UAC MUST include the "histinfo" option tag in the Supported header field in any out-of-dialog requests or initial requests for a dialog for which the UAC would like the History-Info header field in the response. When issuing a request, the UAC MUST follow the procedures in Section 9.2. In the case of an initial request, except where the UAC is part of a B2BUA, there is no cache of hi- entries with which to populate the History-Info header field and the hi-index is set to 1 per Section 10.3. When receiving a response the UAC MUST follow the procedures in Section 9.3.

If the UAC generates further forks of the initial request (either due to acting on a 3xx response or internally-directed forking to multiple destinations), the successive requests will add hi-entries with hi-indexes of 2, 3, etc.

6.2. User Agent Server (UAS) Behavior

When receiving a request, a UAS MUST follow the procedures defined in Section 9.2. When sending a response other than a 3xx response, a UAS MUST follows the procedures as defined in Section 9.4. When sending a 3xx response, the UAS MUST follow the procedures defined for a redirect server per Section 8. An application at the UAS can make use of the cached hi-entries as described in Section 11.

6.3. Back-2-Back User Agent (B2BUA) Behavior

A back-to-back user agent (B2BUA) MAY follow the behavior of a SIP intermediary, per Section 7, as an alternative to following the behavior of a user agent server (UAS) per Section 6.2 and a UAC per Section 6.1. In behaving as an intermediary, a B2BUA carries forward hi-entries received in requests at the UAS to requests being forwarded by the UAC, as well as carrying forward hi-entries in responses received at the UAC to the responses forwarded by the UAS, subject to privacy considerations per Section 10.1.

7. Proxy/Intermediary Handling of History-Info Header Fields

This section describes the procedures for proxies and other SIP intermediaries for the handling of the History-Info header fields for each of the following scenarios:

Receiving a Request: An intermediary MUST follow the procedures in Section 9.1 for the handling of hi-entries in incoming SIP requests.

Sending a Request: For each outgoing request relating to a target in the target set, the intermediary MUST follow the procedures of Section 9.2.

Receiving a Response or Timeout: An intermediary MUST follow the procedures of Section 9.3 when a SIP response is received or a request times out.

Sending a Response: An intermediary MUST follow the procedures of Section 9.4 for the handling of the hi-entries when sending a SIP response.

In some cases, an intermediary may retarget a request more than once before forwarding - i.e., a request is retargeted to a SIP entity that is "internal" to the intermediary before the same intermediary retargets the request to an external target . A typical example would be a proxy that retargets a request first to a different user (i.e., it maps to a different AOR) and then forwards to a registered contact bound to the same AOR. In this case, the intermediary MUST add a hi-entry for (each of) the internal target(s) per the procedures in Section 9.2. The intermediary MAY include a Reason header field in the hi-entry with the hi-targeted-to-uri that has been retargeted. Note, that this is shown in the INVITE (F6) in the example entitled "Sequentially Forking (History-Info in Response)" in [I-D.ietf-sipcore-rfc4244bis-callflows].

8. Redirect Server Handling of History-Info Header Fields

A redirect server MUST follow the procedures in Section 9.1 when it receives a SIP Request. A redirect server MUST follow the procedures in Section 9.4 when it sends a SIP Response. When generating the Contact header field in a 3xx response, the redirect server MUST add the appropriate "mp", "np" or "rc" header field parameter to each Contact header field as described in Section 10.4, if applicable.

9. Handling of History-Info Header Fields in Requests and Responses

This section describes the procedures for SIP entities for the handling of the History-Info header field in SIP requests and responses.

9.1. Receiving a Request

When receiving a request, a SIP entity MUST keep a copy of the hi-entries from the incoming request. This document describes this copy in terms of a cache containing the hi-entries associated with the request. The hi-entries MUST be added to the cache in the order in which they were received in the request.

If the Request-URI of the incoming request does not match the hi-targeted-to-uri in the last hi-entry (i.e., the previous SIP entity that sent the request did not include a History-Info header field), the SIP entity MUST add a hi-entry to end of the cache, on behalf of the previous SIP entity before proceeding to Section 9.2, as follows:

The SIP entity MUST set the hi-targeted-to-uri to the value of the Request-URI in the incoming request. If the Request-URI is a Tel-URI, it SHOULD be transformed into a SIP URI per section 19.1.6 of [RFC3261] before being added as a hi-targted-to-uri.

If privacy is required, the SIP entity MUST follow the procedures of Section 10.1.

The SIP entity MUST set the hi-index parameter as described in Section 10.3.

The SIP entity MUST NOT include an "rc", "mp" or "np" header field parameter.

9.2. Sending a Request with History-Info

When sending a request, a SIP entity MUST include all the hi-entries from the cache that was created per Section 9.1. In addition, the SIP entity MUST add a new hi-entry to the outgoing request, but the SIP entity MUST NOT add the hi-entry to the cache at this time. The hi-entries in the outgoing request's History-Info header field is the preorder of the tree of hi-entries, that is, by the lexicographic ordering of the hi-indexes. The new hi-entry is populated as follows:

hi-targeted-to-uri: The hi-targeted-to-uri MUST be set to the value of the Request-URI of the current (outgoing) request.

privacy: If privacy is required, the procedures of Section 10.1 MUST be followed.

hi-index: The SIP entity MUST include an hi-index for the hi-entry as described in Section 10.3.

rc/mp/np: The SIP entity MUST include an "rc", "mp" or "np" header field parameter in the hi-entry, if applicable, per the procedures in Section 10.4.

9.3. Receiving a Response with History-Info or Request Timeouts

When a SIP entity receives a non-100 response or a request times out, the SIP entity performs the following steps:

Step 1: Add hi-entry to cache

The SIP entity MUST add the hi-entry that was added to the request that received the non-100 response or timed out to the cache, if it was not already cached. The hi-entry MUST be added to the cache in ascending order as indicated by the values in the hi-index parameters of the hi-entries (e.g., 1.2.1 comes after 1.2 but before 1.2.2 or 1.3).

Step 2: Add Reason header field

If the response is not a 100 or 2xx response, the SIP entity adds one or more Reason header fields to the hi-targeted-to-uri in the (newly) cached hi-entry reflecting the SIP response code in the non-100 or non-2xx response, per the procedures of Section 10.2.

Step 3: Add additional hi-entries

The SIP entity MUST also add to the cache any hi-entries received in the response that are not already in the cache. This situation can occur when the entity that generated the non-100 response retargeted the request before generating the response. As per Step 1, the hi-entries MUST be added to the cache in ascending order as indicated by the values in the hi-index parameters of the hi-entries

It is important to note that the cache (and the request or response) does not contain hi-entries for requests that have not yet received a non-100 response, so there can be gaps in indices (e.g., 1.2 and 1.4 could be present but not 1.3).

9.4. Sending History-Info in Responses

When sending a response other than a 100, a SIP entity MUST include all the cached hi-entries in the response, subject to the privacy consideration in Section 10.1.2, and with the following exception: If the received request contained no hi-entries and there is no "histinfo" option tag in the Supported header field, the SIP entity MUST NOT include History-Info in the response.

10. Processing the History-Info Header Field

The following sections describe the procedures for processing the History-Info header field. These procedures are applicable to SIP entities such as Proxies/Intermediaries, Redirect Servers or User Agents.

10.1. Privacy in the History-Info Header Field

The privacy requirements for this document are described in Appendix A.2. Section 10.1.1 describes the insertion of the Privacy header field defined in [RFC3323] to indicate the privacy to be applied to the History-Info header field entries. Section 10.1.2 describes how to apply privacy to a request or response that is being forwarded, based on the presence of the Privacy header field.

10.1.1. Indicating Privacy

As with other SIP headers described in [RFC3323], the hi-targeted-to-uris in the History-Info header field can inadvertently reveal information about the initiator of the request. Thus, the UAC needs a mechanism to indicate that the hi-targeted-to-uris in the hi-entries need to be privacy protected. The Privacy header field is used by the UAC to indicate that privacy is to be applied to all the hi-entries in the request as follows:

- If the UAC is including a Privacy header field with a priv-value of "header" in the request, then the UAC SHOULD NOT include a priv-value of "history" in the Privacy header field in the Request.

- If the UAC is including any priv-values other than "header" in the Privacy header field, then the UAC MUST also include a priv-value of "history" in the Privacy header field in the Request.

- If the UAC is not including any priv-values in the Privacy header field in the request, then the UAC MUST add a Privacy header field, with a priv-value of "history", to the request. The UAC

MUST NOT include a priv-value of "critical" in the Privacy header field in the Request in this case.

In addition, the History-Info header field can reveal general routing and diverting information within an intermediary, which the intermediary wants to privacy protect. In this case, the intermediary MUST construct a Privacy header field with the single priv-value of "history" and include the Privacy header field in the hi-targeted-to-uri, for each new hi-entry created by the intermediary whose hi-targeted-to-uri it wishes to privacy protect. Note that the priv-value in the Privacy header for the incoming request does not necessarily influence whether the intermediary includes a Privacy header field in the hi-entries. For example, even if the Privacy header for the incoming request contained a priv-value of "none", the Proxy can still set a priv-value of "history" in the Privacy header field included in the hi-targeted-to-uri.

Finally, the UAS may not want to reveal the final reached target to the originator. In this case, the UAS MUST include a Privacy header field with a priv-value of "history" in the hi-targeted-to-uri in the last hi-entry, in the response. As noted above, the UAS of the request MUST NOT use any other priv-values in the Privacy header field included in the hi-entry.

10.1.2. Applying Privacy

When a SIP message is forwarded to a domain for which the SIP intermediary is not responsible, a Privacy Service at the boundary of the domain applies the appropriate privacy based on the value of the Privacy header field in the message header or in the "headers" component of the hi-targeted-to-uri in the individual hi-entries.

If there is a Privacy header field in the message header of a request or response, with a priv-value of "header" or "history", then all the hi-targeted-to-uris in the hi-entries, associated with the domain for which the SIP intermediary is responsible, are anonymized by the Privacy Service. The Privacy Service MUST change any hi-targeted-to-uris in these hi-entries that have not been anonymized (evidenced by their domain not being "anonymous.invalid") to anonymous URIs containing a domain of anonymous.invalid as recommended in section 4.1.1.3 of [RFC3323]. As defined in section 4.1.1.2 of [RFC3323] the recommendations of [RFC3261] for anonymyzing the URI Username SHOULD be followed (i.e., "anonymous" in the user portion of the URI). If there is a Privacy header field in the "headers" component of the hi-targeted-to-uri in the hi-entries, then the Privacy header field value MUST be removed from the hi- entry. Once all the appropriate hi-entries have been anonymized, the Privacy Service MUST remove the priv-value of "history" from the Privacy header field in the message

header of the request or response. If there are no remaining priv-values in the Privacy header field, the Privacy Service MUST remove the Privacy header field from the request or response per [RFC3323].

If there is not a Privacy header field in the message header of the request or response that is being forwarded, but there is a Privacy header field with a priv-value of "history" in the "headers" component in any of the hi-targeted-uris in the hi-entries associated with the domain for which a SIP intermediary is responsible, then the Privacy Service MUST update those hi-targeted-to-uris as described above. Any other priv-values in the Privacy header field in the "headers" component of the hi-targeted-to-uris in the hi-entries MUST be ignored. In any case, the Privacy Service MUST remove the Privacy header field from the "headers" compenent of the hi-targeted-to-uris in the hi-entries prior to forwarding.

10.2. Reason in the History-Info Header Field

A Reason header field is added to the "headers" component in an hi-targeted-to-uri when the hi-entry is added to the cache based upon the receipt of a SIP response that is neither a 100 nor a 2xx response, as described in Section 9.3. If the Reason header field is being added due to receipt of an explicit SIP response and the response contains any Reason header fields (see [RFC3326]), then the SIP entity MUST include the Reason header fields in the "headers" component of the hi-targeted-to-uri in the last hi-entry added to the cache, unless the hi-targeted-to-uri is a Tel-URI. If the SIP response does not contain a Reason header field, the SIP entity MUST include a Reason header field, containing the SIP Response Code, in the "headers" component of the hi-targeted-to-uri in the last hi-entry added to the cache, unless the hi-targeted-to-uri is a Tel-URI.

If a request has timed out (instead of being explicitly rejected), the SIP entity MUST update the cache as if the request received a SIP error response code of 408 "Request Timeout".

A request can receive multiple responses, that are neither 100 nor 2xx responses, which carry or imply (for responses without Reason headers, and for timeouts) multiple, possibly duplicated, reason-values to be applied to an hi- targeted-to-uri. In these situations, the SIP entity creating History-Info header value would choose the appropriate Reason header field value.

A SIP entity MAY also include a Reason header field in the "headers" component of an hi-targeted-to-uri containing the URI of a request that was retargeted as a result of internal retargeting.

If additional Reason header field parameters are defined in the

future per [RFC3326], the use of these Reason header field parameters for the History-Info header field MUST follow the same rules as described above.

10.3. Indexing in the History-Info Header Field

In order to maintain ordering and accurately reflect the retargeting of the request, the SIP entity MUST add a hi-index to each hi-entry. Per the syntax in Section 5, the hi-index consists of a series of nonnegative integer separated by dots (e.g., 1.1.2). Each dot reflects a SIP forwarding hop. The nonnegative integer following each dot reflects the order in which a request was retargeted at the hop. The highest nonnegative integer at each hop reflects the number of entities to which the request has been retargeted at the specific hop (i.e., the number of branches) at the time that the request represented by this hi-entry was generated. Thus, the indexing results in a logical tree representation for the history of the request and the hi-entries are given in the preorder of the tree.

The first index in a series of History-Info entries MUST be set to 1. In the case that a SIP entity (intermediary or UAS) adds a first hi-entry on behalf of the previous hop, the hi-index MUST be set to 1. For each forward hop (i.e., each new level of indexing), the last integers of the hi-indexes of the new requests MUST be generated starting at 1 and incrementing by 1 for each additional request.

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

1. Forwarding a request without changing the target: In the case of a request that is being forwarded without changing the target, the hi-index reflects the increasing length of the branch. In this case, the SIP entity MUST read the value from the History-Info header field in the received request and MUST add another level of indexing by appending the dot delimiter followed by an initial value of 1 for the new level. For example, if the hi-index in the last History-Info header field in the received request is 1.1, a proxy would add a hi-entry with an hi-index of 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 SIP entity MUST calculate the hi-index as prescribed for basic forwarding.

3. Retargeting within a processing entity - subsequent instance: For each subsequent retargeting of a request by the same SIP entity, the SIP entity MUST calculate and add the hi-index for each new branch by incrementing the rightmost value from the hi-index in

the last hi-entry. Per the example above, the hi-index in the next request forwarded by this same SIP entity 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 SIP entity MUST calculate the hi-index calculated per rule 3. That is, the rightmost value of the hi-index MUST be incremented (i.e., a new branch is created). For example, if the hi-index in the History-Info header field of the sent request is 1.2 and the response to the request is a 302, then the hi-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 SIP entity MUST set the hi-index for each hi-entry for each forked request per the rules above, with each new request having a unique index. Each index MUST be 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 in the case of parallel forking, only the hi-entry corresponding to the fork is included in the request because no response can yet have been received for any of the parallel forked requests.

6. Missing entry: If the request clearly has a gap in the hi-entry (i.e., the last hi-entry and Request-URI differ), the entity adding an hi-entry MUST add a single index with a value of "0" (i.e., the non-negative integer zero) prior to adding the appropriate index for the action to be taken. For example, if the index of the last hi-entry in the request received was 1.1.2 and there was a missing hi-entry and the request was being forwarded to the next hop, the resulting index will be 1.1.2.0.1. In the case of requests which are forked by a proxy that does not support History-Info, it is possible for hi-entries generated by different entities to have the same index - i.e., each entity supporting History-Info would receive a forked request with the same hi-index to which they would add the value of ".0" prior to adding the appropriate index. Thus, in the previous example, each of the next hop entities would generate an hi-index of 1.1.2.0.1.

10.4. Mechanism for Target Determination in the History-Info Header Field

This specification defines three header field parameters, "rc", "mp" and "np". The header field parameters "rc" and "mp" indicate the mechanism by which a new target for a request is determined. The

header field "np" reflects that the target has not changed. All parameters contain an index whose value is the hi-index of the hi-entry with an hi-targeted-to-uri that represents the Request-URI that was retargeted.

The SIP entity MUST determine the specific parameter field to be included in the hi-target-param, in the History-Info header field, as the targets are added to the target set per the procedures in section 16.5 of [RFC3261] or per section 8.1.3.4 [RFC3261] in the case of retargeting to a contact URI received in a 3xx response. In the latter case, the specific header field parameter in the Contact header field becomes the header field parameter that is used in the hi-entry when the request is retargeted. If the Contact header field does not contain an "rc" or "mp" header field parameter, then the SIP entity MUST NOT include an "rc" or "mp" header field parameter in the hi-target-param in the hi-entry when the request is retargeted to a contact URI received in a 3xx response. This is because the redirect server is the only element with any knowledge on how the target was determined. Note, that the "np" header field parameter is not applicable in the case of redirection.

The SIP entity (intermediary or redirect server) determines the specific header field parameter ("rc", "mp" or "np") to be used based on the following criteria:

- "rc": The Request-URI has changed while retaining the target user associated with the original Request-URI prior to retargeting.

- "mp": The target was determined based on a mapping to a user other than the target user associated with the Request-URI being retargeted.

- "np": The target hasn't changed and the associated Request-URI remained the same.

Note that there are two scenarios by which the "mp" header field parameter can be derived.

- The mapping was done by the receiving entity on its own authority, in which case the mp-value is the parent index of the hi-entry's index.

- The mapping was done due to receiving a 3xx response, in which case the mp-value is an earlier sibling or descendant of an earlier sibling of the hi-entry's index, that of the downstream request which received the 3xx response.

11. Application Considerations

History-Info provides a very flexible building block that can be used by intermediaries and UAs for a variety of services. Prior to any application usage of the History-Info header field parameters, the SIP entity that processes the hi-entries MUST evaluate the hi-entries. The SIP entity MUST be prepared to process effectively messages whose hi-entries show evidence of "gaps", that is, situations that reveal that not all of the forks of the request have been recorded in the hi-entries. Gaps are possible if the request is forwarded through intermediaries that do not support the History-Info header field and are reflected by the existence of hi-entries with a nonnegative integer of "0" e.g. "1.1.0.1". Gaps are also possible in the case of parallel forking if there is an outstanding request at the time the SIP entity sends a message. In addition, gaps may introduce the possibility of duplicate values for the hi-index in the case that a proxy that does not support History-Info forks a request. If gaps are detected, the SIP entity MUST NOT treat this as an error, but SHOULD indicate to any applications that there are gaps. The interpretation of the information in the History-Info header field depends upon the specific application; an application might need to provide special handling in some cases where there are gaps.

The following describes some categories of information that applications can 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. Hi-entry with the index that matches the value of the "rc" header field parameter in the last hi-entry with a "rc" header field parameter in the Request received by a UAS - i.e., the last AOR that was retargeted to a contact based on an AOR-to-contact binding.

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

4. Hi-entry with the index that matches the value of the "rc" header field parameter in the first hi-entry with a "rc" header field parameter in the Request received by a UAS. Note, this would be the original AoR if all the entities involved support the History-Info header field and there is absence of an "mp" header

field parameter prior to the "rc" header field parameter in the hi-target-param in the History-Info header field. However, there is no guarantee that all entities will support History-Info, thus the hi-entry that matches the value of the "rc" header field parameter of the first hi-entry with an "rc" header field parameter in the hi-target-param within the domain associated with the target URI at the destination is more likely to be useful.

5. Hi-entry with the index that matches the value of "mp" header field parameter in the first hi-entry with an "mp" header field parameter in the Request received by a UAS. Note, this would be the original mapped URI if all entities supported the History-Info header field. However, there is no guarantee that all entities will support History-Info, thus the hi-entry that matches the value of the "mp" header field parameter of the first hi-entry with an "mp" header field parameter within the domain associated with the target URI at the destination is more likely to be useful.

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 hi-entry which index matches the value of the "mp" header field parameter of the first hi-entry with an "mp" header field parameter, may also display other agents, reflected by other hi-entries prior to entries with hi-target value of "rc" header field parameter, 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 field is optional, a service MUST define default behavior for requests and responses not containing History-Info header fields. For example, an entity may receive an incomplete set of hi-entries or hi-entries which are not tagged appropriately with an hi-target-param in the case of entries added by entities that are only compliant to RFC4244. 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.

12. Application Specific Usage

The following are possible (non-normative) application-specific usages of History-Info.

12.1. PBX Voicemail

A voicemail system typically requires the original called party information to determine the appropriate mailbox so an appropriate greeting can be provided and the appropriate party notified of the message.

The original target is determined by finding the first hi-entry tagged with "rc" and using the hi-entry referenced by the index of "rc" header field parameter as the target for determining the appropriate mailbox. This hi-entry is used to populate the "target" URI parameter as defined in [RFC4458] The VMS can look at the last hi-entry and find the target of the mailbox by looking at the URI entry in the "target" URI parameter in the hi-entry.

This example usage does not work properly in the presence of forwarding that takes place before the call reaches the company in that case not the first hi-entry with an rc value, but the first hi-entry with an rc value following an mp entry needs to be picked. Further detail for this example can be found in the call flow entitled "PBX Voicemail Example" in [I-D.ietf-sipcore-rfc4244bis-callflows].

Note that in the case where there is no entry tagged with "rc", a VMS can follow the procedures, as defined in [RFC4458], for the "Interaction with Request History Information".

12.2. Consumer Voicemail

The voicemail system in these environment typically requires the last called party information to determine the appropriate mailbox so an appropriate greeting can be provided and the appropriate party notified of the message.

The last target is determined by finding the hi-entry referenced by the index of last hi-entry tagged with "rc" for determining the appropriate mailbox. This hi-entry is used to populate the "target" URI parameter as defined in [RFC4458]. The VMS can look at the last hi-entry and find the target of the mailbox by looking for the "target" URI parameter in the hi-entry. Further detail for this example can be found in the call flow entitled "Consumer Voicemail Example" in [I-D.ietf-sipcore-rfc4244bis-callflows].

In the case where there is no entry tagged with "rc", a VMS can follow the procedures, as defined in [RFC4458], for the "Interaction with Request History Information".

13. Security Considerations

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

This document defines a header field for SIP. The use of the Transport Layer Security (TLS) protocol [RFC5246] as a mechanism to ensure the overall confidentiality of the History-Info header fields (SEC-req-4) is strongly RECOMMENDED. If TLS is NOT used, the intermediary MUST ensure that the messages are only sent within an environment that is secured by other means or that the messages don't leave the intermediary's domain. 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 header fields are no less, nor no more, secure than other SIP header fields, which generally have even more impact on the subsequent processing of SIP sessions than the History-Info header field.

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.

In terms of ensuring the privacy of hi-entries, the same security considerations as those described in [RFC3323] apply. The privacy service that's defined in [RFC3323] MUST also support the new privacy header field priv-value of "history" and anonymize hi-entries in the case of a priv-value of "header" as described in Section 10.1.2.

14. IANA Considerations

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

This document obsoletes [RFC4244] but uses the same SIP header field name, Privacy header field and Option tag. The IANA registry needs to update the references to [RFC4244] with [RFC XXXX], where XXXX is the RFC number for this document.

14.1. Registration of New SIP History-Info Header Field

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

The following row has been updated in the header field section:

Header Name	Compact Form	Reference
History-Info	none	[RFC XXXX]

The following has been updated in the Options Tags section:

Name	Description	Reference
histinfo	When used with the Supported header field, 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.	[RFC XXXX]

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

14.2. Registration of "history" for SIP Privacy Header Field

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

Name	Description	Registrant	Reference
history	Privacy requested for History-Info header fields(s)	Mary Barnes mary.ietf.barnes@gmail.com	[RFC XXXX]

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

14.3. Registration of Header Field Parameters

This specification defines the following new SIP header field parameters in the SIP Header Field parameter sub-registry in the SIP Parameter Registry, http:/www.iana.org/assignments/sip-parameters.

Header Field	Parameter Name	Predefined Values	Reference
History-Info	mp	No	[RFC xxxx]
History-Info	rc	No	[RFC xxxx]
History-Info	np	No	[RFC xxxx]
Contact	mp	No	[RFC xxxx]
Contact	rc	No	[RFC xxxx]
Contact	np	No	[RFC xxxx]

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

15. Acknowledgements

Jonathan Rosenberg et al produced the document that provided additional use cases precipitating the requirement for the new header parameters to capture the method by which a Request URI is determined. The authors would like to acknowledge the constructive feedback provided by Ian Elz, Paul Kyzivat, John Elwell, Hadriel Kaplan, Marianne Mohali, Brett Tate, and Dale Worley. John Elwell also provided excellent suggestions in terms of document structure. Dan Romascanu performed the Gen-ART review.

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.

16. 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 issues.

- [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] gratuitously 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 header field parameters to capture the specific method by which a target is determined to facilitate processing by users of the History-Info header field entries. A specific header field parameter is captured for each of the target URIs as the target set is determined (per section 16.5 of [RFC3261]). The header field parameter is used in both the History-Info and the Contact header fields.

2. Added a way to indicate a gap in History-Info by adding a non-negative integer of "0".

3. Rather than recommending that entries be removed in the case of certain values of the Privacy header field, the entries are anonymized.

4. Updated the security section to be equivalent to the security recommendations for other SIP header fields inserted by intermediaries.

5. Removed Appendix B since a separate call flow document is being published as a companion to this document.

The first 2 changes are intended to facilitate application usage of the History-Info header field 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 and removing the inline references to the requirements. 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 header field parameters.

16.1. Backwards compatibility

This specification is backwards compatible since [RFC4244] allows for the addition of new optional parameters. This specification adds an optional SIP header field parameter to the History-Info and Contact header fields. Entities that have not implemented this specification will ignore these parameters, however, per [RFC4244] an entity will not remove these parameters from an hi-entry. While entities compliant to this document and [RFC4244] must be able to recognize gaps in the hi-entries, this document requires that an index of "0" be used in this case. Whereas [RFC4244] recommended (but did not require) the use of "1". However, since the ABNF in [RFC4244] defines the index as a DIGIT, "0" would be a valid value, thus an [RFC4244] implementation should not have an issue if it receives hi-entries added by intermediaries compliant to this document.

As for the behavior of the UACs, UASs and intermediaries, the following additional normative changes have been made:

UAC behavior

1. Inclusion of option tag by UAC has changed from SHOULD to MUST.

2. Inclusion of hi-target-entry along with hi-index has changed from MAY/RECOMMEND to MUST/MUST.

3. Behavior surrounding the addition of hi-target-entry based on 3xx response has changed from MAY/SHOULD to MUST.

None of the behavior changes would cause any backward or forward compatibility issues.

UAS behavior

1. Inclusion of hi-entry in response has changed from SHOULD to MUST.

As the entity receiving response with hi-entry expected it with SHOULD, this change will not cause any backward compatibility issues.

Proxy/Redirect Server behavior

1. Inclusion of H-I as forwarding the request has changed from SHOULD to MUST.

2. Association of Reason with time-out/internal reason has changed from MAY to MUST.

3. Inclusion of hi-index has changed from RECOMMENDED to MUST.

4. Inclusion of hi-entries in response has changed from SHOULD to MUST.

None of above behavior changes impact backwards compatibility since they only strengthen normative behavior to improve interoperability.

In cases where an entity that is compliant to this document, receives a request that contains hi-entries compliant only to RFC4244 (i.e, the hi-entries do not contain any of the new header field parameters), the entity MUST NOT add any of the new header field parameters to the hi-entries. The hi-entries MUST be cached and forwarded as any other entries are as specified in Section 9.1. As with RFC4244 compliant entities, applications must be able to function in cases of missing information, as specified in Section 11.

17. References

17.1. 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.

[RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[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.

17.2. 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.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[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.

[I-D.ietf-sipcore-rfc4244bis-callflows]
Barnes, M., Audet, F., Schubert, S., Elburg, H., and C. Holmberg, "Session Initiation Protocol (SIP) History-Info Header Call Flow Examples", draft-ietf-sipcore-rfc4244bis-callflows-06 (work in progress), July 2013.

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 MUST also generate Request History information for the 'internal retargeting'.

   B. An entity (UA or proxy) retargeting in response to a redirect or REFER MUST 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 whether 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 early or 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-Info entry either by adding an additional hi-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 out going messages unless it is protected as described in [RFC3323].

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@ntt-at.com

Hans Erik van Elburg
Detecon International Gmbh
Sternengasse 14-16
Cologne,
Germany

Email: ietf.hanserik@gmail.com

Christer Holmberg
Ericsson
Hirsalantie 11, Jorvas
Finland

Email: christer.holmberg@ericsson.com