

TRAW Working Group
Internet Draft
Intended status: Informational
Expires: August 25, 2013

H. Kaplan
Acme Packet
February 25, 2013

A Taxonomy of Session Initiation Protocol (SIP)
Back-to-Back User Agents
draft-ietf-straw-b2bua-taxonomy-01

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 25, 2013.

Copyright Notice

Copyright (c) 2012 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.

In many SIP deployments, SIP entities exist in the SIP signaling path between the originating UAC and final terminating UAS, which go beyond the definition of a Proxy, performing functions not defined in standards-track RFCs. The only term for such devices provided in [RFC3261] is for a Back-to-Back User Agent (B2BUA), which is defined as the logical concatenation of a User Agent Server (UAS) and User Agent Client (UAC).

There are numerous types of SIP Back-to-Back User Agents (B2BUAs), performing different roles in different ways. For Example IP-PBXs, SBCs and Application Servers. This document identifies several common B2BUA roles, in order to provide taxonomy other documents can use and reference.

Table of Contents

| | |
|---|---|
| 1. Terminology..... | 3 |
| 2. Introduction..... | 3 |
| 3. B2BUA Role Types..... | 3 |
| 3.1. Signaling-plane B2BUA Roles..... | 4 |
| 3.1.1 Proxy-B2BUA | 4 |
| 3.1.2 Signaling-only | 4 |
| 3.1.3 SDP-Modifying Signaling-only | 4 |
| 3.2. Media-plane B2BUA Roles..... | 5 |
| 3.2.1 Media-relay | 5 |
| 3.2.2 Media-aware | 5 |
| 3.2.3 Media-termination | 6 |
| 4. Mapping SIP Device Types to B2BUA Roles..... | 6 |
| 4.1. SIP PBXs and Softswitches..... | 6 |
| 4.2. Application Servers..... | 6 |
| 4.3. Session Border Controllers..... | 6 |
| 4.4. Transcoders..... | 7 |
| 4.5. Conference Servers..... | 7 |
| 4.6. P-CSCF and IBCF Functions..... | 7 |
| 4.7. S-CSCF Function..... | 8 |
| 5. Security Considerations..... | 8 |
| 6. IANA Considerations..... | 8 |
| 7. Acknowledgments..... | 8 |
| 8. References..... | 8 |
| 8.1. Informative References..... | 8 |
| Author's Address..... | 9 |

1. Terminology

B2BUA: a SIP Back-to-Back User Agent, which is the logical combination of a User Agent Server (UAS) and User Agent Client (UAC).

UAS: a SIP User Agent Server.

UAC: a SIP User Agent Client.

2. Introduction

In current SIP deployments, there are numerous forms of B2BUAs, operating at various layers of the protocol stack, and for various purposes, and with widely varying behavior from a SIP protocol perspective. Some act as pure SIP Proxies and only change to the role of B2BUA in order to generate BYEs to terminate dead sessions. Some are full User Agent stacks with only high-level event and application logic binding the UAS and UAC sides. Some B2BUAs operate only in the SIP signaling plane, while others participate in the media plane as well.

As more and more SIP domains get deployed and interconnect the probability of a single SIP session crossing multiple B2BUA's at both the signaling and media planes increases significantly.

This document provides a taxonomy of several common B2BUA roles, so that other documents may refer to them using their given names without re-defining them in each document.

3. B2BUA Role Types

Within the context of this document, the terms refer to a B2BUA role, not a particular system type. A given system type may change its role in the middle of a SIP session, for example when a Stateful Proxy tears-down a session by generating BYEs; or for example when an SBC performs transcoding or REFER termination.

Furthermore, this document defines 'B2BUA' following the definition provided in [RFC3261], which is the logical concatenation of a UAS and UAC. A typical centralized conference server, for example, is not a B2BUA because it is the target UAS of multiple UACs, whereby the UACs individually and independently initiate separate SIP sessions to the central conference server. Likewise, a third-party call control transcoder as described in section 3.1 of [RFC5369] is not a B2BUA; whereas an inline transcoder based on [RFC5370] is a B2BUA.

3.1. Signaling-plane B2BUA Roles

A Signaling-plane B2BUA is one that operates only on the SIP message protocol layer, and only with SIP messages and headers but not the media itself in any way. This implies it does not modify SDP bodies, although it may save them and/or operate on other MIME body types. This category is further subdivided into specific roles as described in this section.

3.1.1 Proxy-B2BUA

A Proxy-B2BUA is one that appears, from a SIP protocol perspective, to be a SIP Proxy based on [RFC3261] and its extensions, except that it maintains sufficient dialog state to generate in-dialog SIP messages on its own and does so in specific cases. The most common example of this is a SIP Proxy which can generate BYE requests to tear-down a dead session.

A Proxy-B2BUA does not modify the received header fields such as the To, From, or Contact, and only modifies the Via and Record-Route header fields following the rules in [RFC3261] and its extensions. If a Proxy-B2BUA can generate in-dialog messages, then it will also need to modify the CSeq header, after it's generated its own. A Proxy-B2BUA neither modifies nor inspects MIME bodies (including SDP), does not have any awareness of media, will forward any Method type, etc.

3.1.2 Signaling-only

A Signaling-only B2BUA is one that operates at the SIP layer but in ways beyond those of [RFC3261] Proxies, even for normally forwarded requests. Such a B2BUA may or may not replace the Contact URI, modify or remove all Via and Record-Route headers, modify the To and From header fields, modify or inspect specific MIME bodies, etc. No SIP header field is guaranteed to be copied from the received request on the UAS side to the generated request on the UAC side.

An example of such a B2BUA would be some forms of Application Servers and PBXs, such as a server which locally processes REFER requests and generates new INVITES on behalf of the REFER's target. Another example would be an [RFC3323] Privacy Service Proxy performing the 'header' privacy function.

3.1.3 SDP-Modifying Signaling-only

An SDP-Modifying Signaling-only B2BUA is one that operates in the signaling plane only and is not in the media path, but does modify SDP bodies and is thus aware of and understands SDP syntax and

semantics. Some Application Servers and PBXs act in this role in some cases, for example to remove certain codec choices or merge two media endpoints into one SDP offer.

3.2. Media-plane B2BUA Roles

A Media-plane B2BUA is one that operates at both the SIP and media planes, not only on SIP messages but also SDP and potentially RTP/RTCP or other media. Such a B2BUA may or may not replace the Contact URI, modify or remove all Via and Record-Route headers, modify the To and From header fields, etc. No SIP header field is guaranteed to be copied from the received request on the UAS side to the generated request on the UAC side, and SDP will also be modified.

An example of such a B2BUA would be a Session Border Controller performing the functions defined in [RFC5853], a B2BUA transcoder as defined in [RFC5370], a rich-ringtone Application Server, or a recording system. Another example would be an [RFC3323] Privacy Service Proxy performing the 'session' privacy function.

Note that a Media-plane B2BUA need not be instantiated in a single physical system, but may be decomposed into separate signaling and media functions.

The Media-plane B2BUA category is further subdivided into specific roles as described in this section.

3.2.1 Media-relay

A B2BUA which performs a media-relay role is one that terminates the media plane at the IP and UDP/TCP layers on its UAS and UAC sides, but neither modifies nor restricts which forms of UDP or TCP payload are carried within the UDP or TCP packets. Such a role may only support UDP or only TCP or both, as well as other transport types or not. Such a role may involve policing the IP packets to fit within a bandwidth limit, or converting from IPv4 to IPv6 or vice-versa. This is typically similar to a NAT behavior, except a NAT operating in both directions on both the source and destination information; it is often found as the default behavior in SBCs.

3.2.2 Media-aware

A B2BUA which performs a media-aware role is similar to a media-relay except that it inspects and potentially modifies the payload carried in UDP or TCP, such as [RFC3550] RTP or RTCP, but not at a codec or higher layer. An example of such a role is an [RFC3711] SRTP terminator, which does not need to care about the RTP payload but does care about the RTP header; or a device which monitors RTCP

for QoS information; or a device which muxes/de-muxes RTP and RTCP packets on the same 5-tuple.

3.2.3 Media-termination

A B2BUA which performs a media-termination role is one that operates at the media payload layer, such as RTP/RTCP codec or MSRP message layer and higher. Such a role may only terminate/generate specific RTP media, such as [RFC4733] DTMF packets, or it may convert between media codecs, or act as a Back-To-Back [RFC4975] MSRP agent. This is the role performed by transcoders, conference servers, etc.

4. Mapping SIP Device Types to B2BUA Roles

Although the B2BUA role types defined previously do not define a system type, as discussed in section 3, some discussion of what common system types perform which defined roles may be appropriate. This section provides such a 'mapping' for general cases, to aid in understanding of the roles.

4.1. SIP PBXs and Softswitches

SIP-enabled Private Branch eXchanges (SIP PBXs) and Softswitches are market category terms, and not specified in any standard. In general they can perform every role described in this document at any given time, based on their architecture or local policy. Some are based on architectures that make them the equivalent of a SIP UAS and UAC connected with a logical PRI loopback; others are built as a SIP Proxy core with optional modules to "do more".

4.2. Application Servers

Application Servers are a broad marketing term, and not specified in any standard in general, although 3GPP and other organizations specify some specific Application Server functions and behaviors. Common examples of Application Servers functions are message-waiting indication, find-me-follow-me services, privacy services, call-center ACD services, call screening, and VCC services. Some only operate in the signaling plane in either Proxy-B2BUA or Signaling-only B2BUA roles; others operate as full Media-termination B2BUAs, such as when providing IVR, rich-ringtone or integrated voicemail services.

4.3. Session Border Controllers

Session Border Controllers (SBCs) are a market category term, and not specified in any standard. Some of the common functions performed by SBCs are described in [RFC5853], but in general they

can perform every role described in this document at any given time, based on local policy. By default, most SBCs are either Media-relay or Media-aware B2BUAs, and replace the Contact URI, remove the Via and Record-Route headers, modify the Call-ID, To, From, and various other headers, and modify SDP. Some SBCs remove all headers, all bodies, and reject all Method types unless explicitly allowed by local policy; other SBCs pass all such elements through unless explicitly forbidden by local policy.

4.4. Transcoders

Transcoders perform the function of transcoding one audio or video media codec type to another, such as G.711 to G.729. As such they perform the Media-termination role, although they may only terminate media in specific cases of codec mismatch between the two ends. Although [RFC5369] and [RFC5370] define two types of SIP Transcoders, in practice a popular variant of the [RFC5370] inline model is to behave as a SIP B2BUA without using the resource-list mechanism, but rather simply route a normal INVITE request through an inline transcoder. SIP Transcoders architectures are based on everything from SIP media servers, to SBCs, to looped-back TDM gateways, and thus run the gamut from replacing only specific headers/bodies and SDP content needed to perform their function, to replacing almost all SIP headers and SDP content. Some transcoders save and remove SDP offers in INVITEs from the UAC, and wait for an offer in the response from the UAS, similar to a 3PCC model; others just insert additional codecs in SDP offers and only transcode if the inserted codec(s) are selected in the answer.

4.5. Conference Servers

In general Conference Servers do not fall under the term 'B2BUA' as defined by this document, since typically they involve multiple SIP UACs initiating independent SIP sessions to the single conference server UAS. However, a conference server supporting [RFC5366], whereby the received INVITE triggers the conference focus UAS to initiate multiple INVITEs as a UAC, would be in a Media-termination B2BUA role when performing that function.

4.6. P-CSCF and IBCF Functions

Proxy-CSCFs and IBCFs are functions defined by [3GPP] IMS standards, and when coupled with the IMS media-plane gateways (IMS AGW, TrGW, etc.) typically form a logical Media-relay or Media-aware B2BUA role.

4.7. S-CSCF Function

S-CSCF is a function defined by [3GPP] IMS standards, and typically follows a Proxy-B2BUA role.

5. Security Considerations

The security considerations related to the functionality described in this document are addressed in the relevant references.

6. IANA Considerations

This document makes no request of IANA.

7. Acknowledgments

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).

8. References

8.1. Informative 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.

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

[RFC3550] Schulzrinne, H., et al, "RTP: A Transport Protocol for Real-Time Applications", RFC 3550, July 2003.

[RFC3711] Baugher, M., et al, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[RFC4733] Schulzrinne, H., Taylor, T., "RTP Payload for DTMF Digits, Telephony Tones, and Telephony Signals", RFC 4733, December 2006.

[RFC4975] Campbell, B., Mahy, R., Jennings, C., "The Message Session Relay Protocol (MSRP)", RFC 4975, September 2007.

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

[RFC5369] Camarillo, G., "Framework for Transcoding with the Session Initiation Protocol (SIP)", RFC 5369, October 2008.

Internet-Draft Taxonomy of B2BUAs February 2013
[RFC5370] Camarillo, G., "The Session Initiation Protocol (SIP)
Conference Bridge Transcoding Model", RFC 5370, October 2008.

[RFC5853] Hautakorpi, J, et al, "Requirements from Session
Initiation Protocol (SIP) Session Border Control (SBC) Deployments",
RFC 5853, April 2010.

[3GPP] 3GPP, "IP Multimedia Subsystem (IMS); Stage 2", 3GPP TS
23.228.

Author's Address

Hadriel Kaplan
Acme Packet
Email: hkaplan@acmepacket.com

Network Working Group
Internet Draft
Intended status: Standards Track
Expires: August 30, 2013

H. Kaplan
Acme Packet
Victor Pascual
Acme Packet
February 25, 2013

Loop Detection Mechanisms for
Session Initiation Protocol (SIP)
Back-to-Back User Agents (B2BUAs)
draft-kaplan-straw-b2bua-loop-detection-01

Abstract

SIP Back-to-Back User Agents (B2BUAs) can cause unending SIP request routing loops because, as User Agent Clients, they can generate SIP requests with new Max-Forwards values. This document discusses the difficulties associated with loop detection for B2BUAs, and requirements for them to prevent infinite loops.

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on August 25, 2013.

Copyright Notice

Copyright (c) 2012 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.

Table of Contents

| | |
|---------------------------------------|---|
| 1. Terminology..... | 2 |
| 2. Introduction..... | 2 |
| 3. Background..... | 3 |
| 4. B2BUA Loop-Detection Behavior..... | 4 |
| 5. B2BUA Max-Forwards Behavior..... | 4 |
| 6. B2BUA Max-Breadth Behavior..... | 4 |
| 7. Security Considerations..... | 5 |
| 8. IANA Considerations..... | 5 |
| 9. Acknowledgments..... | 5 |
| 10. References..... | 5 |
| 10.1. Informative References..... | 5 |
| Authors' Addresses..... | 6 |

1. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119. The terminology in this document conforms to RFC 2828, "Internet Security Glossary".

B2BUA terminology and taxonomy used in this document is based on [draft-b2bua-taxonomy].

2. Introduction

SIP provides a means of preventing infinite request forwarding loops in [RFC3261], and a means of mitigating parallel forking amplification floods in [RFC5393]. Neither document normatively defines specific behavior for B2BUAs, however.

Unbounded SIP request loops have actually occurred in SIP deployments, numerous times. The cause of loops is usually mis-configuration, but the reason they have been unbounded/unending is they crossed B2BUAs that reset the Max-Forwards value in the SIP

requests they generated on their UAC side. Although such behavior is technically legal per [RFC3261] because a B2BUA is a UAC, the resulting unbounded loops have caused service outages and make troubleshooting difficult.

Furthermore, [RFC5393] also provides a mechanism to mitigate the impact of parallel forking amplification issues, through the use of a "Max-Breadth" header field. If a B2BUA does not pass on this header field, parallel forking amplification is not mitigated with the [RFC5393] mechanism.

This document defines normative requirements for Max-Forwards and Max-Breadth header field behaviors of B2BUAs, in order to mitigate the effect of loops and parallel forking amplification.

3. Background

Within the context of B2BUAs, the scope of the SIP protocol ends at the UAS side of the B2BUA, and a new one begins on the UAC side. A B2BUA is thus capable of choosing what it wishes to do on its UAC side independently of its UAS side, and still remain compliant to [RFC3261] and its extensions. For example, any B2BUA type defined in [draft-b2bua-taxonomy] other than Proxy-B2BUA may create the SIP request on its UAC side without copying any of the Via header field values received on its UAS side. Indeed there are valid reasons for it to do so; however this prevents the Via-based loop-detection mechanism defined in [RFC3261] and updated by [RFC5393] from detecting SIP request loops any earlier than by reaching a Max-Forwards limit.

Some attempts have been made by B2BUA vendors to detect request loops in other ways: by keeping track of the number of outstanding dialog-forming requests for a given caller/called URI pair; or by detecting when they receive and send their own media addressing information too many times in certain cases when they are a Media-plane B2BUA; or by encoding a request instance identifier in some field they believe will pass through other nodes, and detecting when they see the same value too many times.

All of these methods are brittle and prone to error, however. They are brittle because the definition of when a value has been seen "too many times" is very hard to accurately determine; requests can and do fork before and after B2BUAs process them, and requests legitimately spiral in some cases, leading to incorrect determination of loops. The mechanisms are prone to error because there can be other B2BUAs in the loop's path that interfere with the particular mechanism being used.

Ultimately, the last defense against loops becoming unbounded is to limit how many SIP hops any request can traverse, which is the purpose of the SIP Max-Forwards field value. If B2BUAs were to at least copy and decrement the Max-Forwards header field value from their UAS to the UAC side, loops would not continue indefinitely.

4. B2BUA Loop-Detection Behavior

A Proxy-B2BUA, as defined in [draft-b2bua-taxonomy], MUST implement the loop-detection mechanism for the Via header field, as defined for a Proxy in [RFC5393].

[Note: should we require all B2BUAs to perform Via-header loop-detection as well, even if they themselves don't forward on the Via headers?]

5. B2BUA Max-Forwards Behavior

All B2BUA types MUST copy the received Max-Forwards header field from the received SIP request on their UAS side, to any request(s) they generate on their UAC side, and decrement the value, as if they were a Proxy following [RFC3261].

Being a UAS, B2BUAs MUST also check the received Max-Forwards header field and reject or respond to the request if the value is zero, as defined in [RFC3261].

If the received request did not contain a Max-Forwards header field, one MUST be created in any requests generated in the UAC side, which SHOULD be 70, as described for Proxies in section 16.6 part 3 of [RFC3261].

For B2BUAs that remove Record-Route headers, they MUST only perform the copying and checking rules above for out-of-dialog requests. The reason for this is other User Agents might send in-dialog requests using a very low Max-Forwards value, based on the number of Record-Route headers they received.

6. B2BUA Max-Breadth Behavior

All B2BUA types MUST copy the received Max-Breadth header field from the received SIP request on their UAS side, to any request(s) they generate on their UAC side, as if they were a Proxy following [RFC5393].

B2BUAs of all types MUST follow the requirements imposed on Proxies as described in section 5.3.3 of [RFC5393], including generating the header field if none is received, limiting its maximum value, etc.

Internet-Draft Loop Detection for B2BUAs February 2013
B2BUAs that generate parallel requests on their UAC side for a single incoming request on the UAS side MUST also follow the rules for Max-Breadth handling in [RFC5393] as if they were a parallel forking Proxy.

7. Security Considerations

The security implications for parallel forking amplification are documented in section 7 of [RFC5393]. This document does not add any additional issues beyond those discussed in [RFC5393].

Some B2BUAs reset the Max-Forwards and Max-Breadth header field values in order to obfuscate the number of hops a request has already traversed, as a privacy or security concern. Such goals are at odds with the mechanisms in this document, and administrators can decide which they consider more important: obfuscation vs. loop detection. In order to comply with this RFC, manufacturers MUST comply with the normative rules defined herein by default, but MAY provide user-configurable overrides as they see fit.

8. IANA Considerations

This document makes no request of IANA.

9. Acknowledgments

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA). Thanks to Brett Tate for his review of the document.

10. References

10.1. Informative 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.
- [RFC5393] Sparks, R., et al, "Addressing an Amplification Vulnerability in Session Initiation Protocol (SIP) Forking Proxies", RFC 5393, December 2008.
- [draft-b2bua-taxonomy] Kaplan, H., "A Taxonomy of Session Initiation Protocol (SIP) Back-to-Back User Agents", draft-kaplan-straw-b2bua-taxonomy-00, July 30, 2012.

Internet-Draft
Authors' Addresses

Loop Detection for B2BUAs

February 2013

Hadriel Kaplan
Acme Packet
Email: hkaplan@acmepacket.com

Victor Pascual
Acme Packet
Email: vpascual@acmepacket.com