Abstract

There are several reasons for having a globally unique session identifier for the same SIP session, which can be maintained across B2BUA’s and other SIP middle-boxes. This draft proposes a new SIP header to carry such a value: Session-ID.
1. Introduction

The SIP [RFC3261] Call-ID header value is a globally unique identifier, mandatory in all requests/responses, which identifies SIP messages belonging to the same dialog or registration. It provides a portion of the SIP message dialog-matching criteria, and is used in such things as [Replaces] headers and [dialog-events] package for matching to dialogs, and in [SIP-Identity] and [Connected-identity] as one of the inputs for signing.

Unfortunately, the Call-ID is often changed by B2BUA’s and other such middle-boxes in the end-to-end message path. A B2BUA logically represents a UAS and UAC, and as such may use a new Call-ID value for the dialog it creates on its UAC half; but there are several use-cases for having a common, consistent end-to-end identifier, as described later in this draft.
There are several reasons the Call-ID value is changed by B2BUA’s. There are security and privacy reasons, since Call-ID values typically contain UA IP Addresses; some B2BUA’s need to change them to keep track of spiraling dialogs; and some need to change them to keep track of separate forks. In fact, some people have argued a B2BUA has no choice but to create a new one, in order to strictly comply with RFC 3261 as a UAC. In general, B2BUA’s modify the Call-ID value in both directions, "fixing" it to be what each side of the B2BUA would expect. This works fine if the B2BUA is in the message path, and knows all SIP message or body contents which use or reference the value. However for subsequent out-of-dialog requests, or new SIP uses, a B2BUA often does not or cannot "fix" the value correctly, for example if it is not traversed.

Therefore, in order to provide an identifier which will not be modified/replaced by B2BUA’s, this draft proposes a new SIP Header "Session-ID", and mandatory rules for the value of such a header. The rules are designed to be such that the value in the Session-ID header is not considered unsafe, private, or have any property which would cause B2BUA’s to change it. The goal of this draft is to enable use-cases which need a unique identifier for a given session which can successfully cross B2BUA’s.

1.1. Requirements

The following requirements drive the need for Session-ID:

REQ1: It must be possible to identify a set of dialogs which have a direct correlation with each other such that they represent the same SIP session, with as high a probability as possible.

REQ2: It must be possible to pass the identifier through B2BUA’s, with as high a probability as possible. This requirement drives the following requirements:

REQ2a: The identifier must not reveal any information related to any SIP device or domain identity, including IP Address, port, hostname, domain name, username, Address-of-Record, MAC address, IP address family, transport type, etc.

REQ2b: The identifier must not reveal to the receiver of it that the Call-ID, tags, or any other SIP header or body portion have been changed by middle-boxes, with as high a probability as possible.

In a previous version of this draft, an additional requirement was proposed that the identifier be usable in out-of-dialog requests for matching purposes, similar to the Call-ID usage today. The motivation for this was to enable SIP use-cases to work that
currently do not work, or may not work in the future as SIP domains connect together. Concerns were raised that the solution would not solve all possible use-cases, in all possible scenarios, and thus would not be a complete solution to the problem. Therefore, in order to progress this draft for troubleshooting uses, that additional requirement and its associated solution behavior has been removed from this draft.

1.2. The use-case for Session-ID

The need for a unique identifier is driven by the need to troubleshoot SIP sessions as they cross SIP nodes. Troubleshooting is more complicated if multiple legs of the session are on different sides of B2BUA’s, due to the lack of a common identifier to tie the legs together. Currently proprietary mechanisms are used to achieve this.

2. 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".

3. Applicability

This draft proposes a new SIP header for all requests and responses.

4. Overview of Operation

The general concept is that the UAC generating an out-of-dialog request generates a new, pseudo-random, unique value which remains constant for the duration of the transaction, any dialog created from the request, or a registration. The value is based on the rules for creating a fixed-length pseudo-random value, and is inserted in a new Session-ID header defined in this draft. The UAC and UAS then reflect this value in all messages for the duration of the dialog.

To aid in migration of deployments, a B2BUA or Proxy may also generate and/or insert the value on behalf of a UAC or UAS, if one or the other does not support this document’s mechanism.

This Session-ID is not used for message dialog-matching rules in RFC 3261, nor does it change the Call-ID usage, nor does it replace the Call-ID value. Instead this new header value provides an identifier for troubleshooting uses only.
5. Session-ID Behavior

5.1. Generating a Session-ID value

This draft proposes the Session-ID header value be generated based on a defined hash mechanism for creating a 128-bit pseudo-random value, and encode it as its lower-case hex representation. The reason for specifying the mechanism, and its length, is to make it impossible to determine the manufacturer of the device which generated it by looking at its format or value. For example, the theoretically random "session id" value in SDP origin line has been found to be fairly vendor-specific in nature, and one can narrow the vendor that generated the SDP simply by the origin session id value. (In fact, this drove some SBC's to modify that SDP field for "anonymization" purposes)

In order to enable trouble-shooting of in-dialog messages, a generator needs to remember the Session-ID for a given dialog(s). This is described in more detail in following sections of this document.

The Session-ID value is generated by taking the Call-ID header value, and SHA-1 hashing it based on [RFC2104] HMAC using a locally generated pseudo-random 128-bit system secret key, to create a 128-bit resultant HMAC value. The secret key makes the resultant HMAC value not re-creatable by other parties, which is necessary to prevent detection of Call-ID’s being changed, as required by Req-3b. Otherwise, middle-boxes may have motivation to remove the Session-ID in order to hide the fact that they changed the Call-ID.

Per [RFC2104], the algorithm is thus HMAC-SHA-1-128(Call-ID_value, secret_key), and the 128-bit result is encoded using lowercase alphanumeric hex representation, as defined in the ABNF section of this document.

5.2. UAC Behavior

The rules for when a UAC generates a new Session-ID value are similar as those for Call-ID value: a UAC supporting this draft MUST generate a *new* unique Session-ID value whenever it generates an out-of-dialog request, or for a new Registration. The UAC MUST re-use the same Session-ID for in-dialog messages, and for any out-of-dialog request it retransmits or re-generates in response to a 3xx, or it re-formulates due to failure responses. This follows the rules in [RFC3261] for Call-ID generation.

Session-ID values in Registration "refreshes" - REGISTER requests which are used to update the expiry time but not to register a new
contact - MUST use the same Session-ID value as previous REGISTER requests. New Registrations, which add or change the Contact URI for the AoR, but not simply to delete them, MUST use a new Session-ID value. This follows the behavior of Call-ID per RFC 3261 and thus the hash mechanism should by definition produce the correct Session-ID; it is re-iterated here because some devices incorrectly change their Call-ID value for every re-Registration, and MUST NOT do the same to the Session-ID.

The UAC MUST include the Session-ID header value in every SIP message it transmits. This serves both a troubleshooting purpose, and may be used in specific identity verification mechanisms which are beyond the scope of this draft.

5.3. UAS Behavior

A UAS compliant with this document MUST copy a received Session-ID value in a request, into responses and subsequent upstream requests sent within the dialog.

If an out-of-dialog request is received without a Session-ID header field, the UAS SHOULD generate a new one for subsequent use in the transaction and dialog, as defined for a UAC, and use the value in all responses and upstream in-dialog requests.

5.4. Proxy Behavior

A Proxy MUST NOT remove or modify the Session-ID header values it receives, if one is in the message. By definition, an RFC 3261 compliant Proxy would not modify or remove such a header.

A Proxy compliant with this draft MAY generate a new Session-ID or insert a previously saved one, if and only if none existed in a message, following the rules for doing so as a B2BUA defined later.

If the Proxy forks a request, it MUST copy the same Session-ID value into all the forked request copies. If the Proxy recourses requests due to 3xx redirection, or regenerates requests due to failures, it MUST use the same Session-ID value as the original request, just as the UAC does.

If the Proxy locally generates any response or request based on a received request, including 100 Trying, it MUST insert any received Session-ID value from the original request into the response message it locally creates. This is necessary for troubleshooting purposes.

5.5. B2BUA Behavior
A B2BUA compliant with this document MUST copy the Session-ID it receives in requests as a UAS into the related requests it generates as a UAC; and any Session-ID value it receives in responses as a UAC into the correlated responses it generates as a UAS.

If the B2BUA forks or creates multiple requests as a UAC, from a request it received as a UAS, the B2BUA MUST copy the same Session-ID header value it received into all the forks/requests. If the B2BUA recurses requests due to 3xx redirection, or regenerates requests due to failures, it MUST use the same Session-ID value, just as the UAC does.

If the B2BUA locally generates any response or request based on a received request, including 100 Trying, it MUST insert any received Session-ID value from the original request into the response message it locally creates. A B2BUA MUST remember the received Session-ID value for the duration of the transaction and dialog, for the purpose of re-insertion, in case the far-end does not support this draft.

In all cases, if the SIP message received by a B2BUA contained a Session-ID header field, a B2BUA compliant with this document MUST NOT remove, modify or replace the header value.

5.5.1 B2BUA Generation of New Session-ID

If an out-of-dialog request is received by a B2BUA compliant with this document, and the request does *not* contain a Session-ID header field, the B2BUA MUST generate a new Session-ID. It MUST then insert it in any requests or responses it generates, as if it had actually received the new Session-ID from the UAC, following the rules previously defined for a B2BUA. This allows for a B2BUA to provide a migration to Session-ID deployment, on behalf of upstream nodes that do not yet support it. As defined previously, if any received message already had a Session-ID, a B2BUA compliant with this document would not replace it.

5.5.2 B2BUA Insertion of Saved Session-ID

If a Session-ID was received in an out-of-dialog request, or the B2BUA locally generated one because none existed, the B2BUA SHOULD insert the same Session-ID value into all responses and upstream in-dialog requests if and only if a Session-ID is not already in them. This allows for a B2BUA to provide a migration to Session-ID deployment, on behalf of downstream nodes that do not yet support it.
6. Session-ID Migration and Failure Scenarios

SIP is already widely deployed on the Internet, and it is impractical to expect all UA’s to be upgraded to support this document’s mechanism in the near future. A solution for gradual migration is necessary, which this document provides by allowing B2BUA’s or Proxies to perform the Session-ID generator and inserter role. Even within those device types, it is impractical to expect all B2BUA’s to support this mechanism all at once, or any time in the near future. Therefore, it is expected that some B2BUA’s and/or UA’s will support generating and inserting Session-ID, while others will not support Session-ID at all.

Due to the varying types of B2BUAs, such as SBCs, Application Servers, Feature Servers, and Softswitches of various flavors, and the numerous SIP deployment models in use, there are going to be cases in which Session-ID will fail to be a consistent value for all related dialogs, or fail to successfully match. The goal of this draft is to improve current deployments as much as possible - not to cover all possible scenarios - and in this author’s opinion that is the best that can be done given the constraints.

One example is for forked requests: if a UAC which does not support this mechanism sends a request to a Proxy or B2BUA which also does not support this mechanism, each fork could reach B2BUA’s or UAS’s which *do* support this mechanism. In such a case, each of those forked-to B2BUA/UAS will generate unique Session-IDs and put them in their responses, leading to two different Session-ID values for the same related dialogs temporarily. Eventually the UAC would only accept one of the dialogs (typically), and only one Session-ID would remain.

7. New Header

The following table updates Table 2 in [RFC3261] and other defined extensions.

```
Hdr-field when ACK BYE CAN INV OPT REG PRA INF REF UPD SUB NOT MSG
Session-ID  R   m   m   m   m   m   m   m   m   m   m   m   m   m
Session-ID  r   m   m   m   m   m   m   m   m   m   m   m   m   m
```

7.1. "Session-ID" header

This draft proposes "Session-ID" to be added to the definition of the element "message-header" in the SIP message grammar.
The Session-ID header is a single-instance header.

The compact form of the header is requested to be: h
[for "Help us help you"]

7.2. Augmented BNF Definitions

```
Session-ID = "Session-ID" HCOLON sess-id *( SEMI generic-param )
  sess-id = 32(DIGIT / %x61-7A) ; 32 chars of [0-9a-z]
```

NOTE: The sess-id value is case-SENSITIVE, just as Call-ID is, however only lower-case characters are defined and allowed.

See the Security Considerations section for discussion about using header parameters in Session-ID header fields.

8. Example Exchange

In the following example, Alice initiates a call to Bob. Alice generates a Session-ID header in the out-of-dialog INVITE.

Alice generates the following: (note: much has been left out for simplicity)
```
INVITE sip:bob@example.com SIP/2.0
Via: SIP/2.0/UDP 192.0.2.1:5060;branch=z9hG4bKnashds10
From: Alice <sip:alice@example.net>;tag=1234567
To: Bob <sip:bob@example.com>
Call-Id: 123456mcmxcix@1.2.3.4
Session-ID: f81d4fae7dec11d0a76500a0c91e6bf6
CSeq: 1 INVITE
Contact: <sip:alice@192.168.1.1>
```

9. Security Considerations

There are several security considerations surrounding this document’s mechanism.

The Session-ID’s value is created from the Call-ID using a hashing mechanism based on [RFC2104], using SHA-1 and a secret key known only to the system generating the Session-ID. Because the algorithm is defined in this document, it should be fairly secure from detecting the generator of the Session-ID, in terms of manufacturer or code base.

The Session-ID generation algorithm should provide a reasonably random 128-bit Session-ID value, to avoid collisions, and would not let one re-create the original Call-ID. The secret key MUST only be
used for the Session-ID mechanism, in case a weakness is found which reveals the key. One such weakness may be that a UAC generates one or more Call-ID’s which have a property that makes determining the key more likely.

9.1. Security considerations for B2BUA vendors and operators

The requirement for the Session-ID is to be an identifier which cannot be used by a recipient to identify if the Call-ID has been changed by middle-boxes. As such, a UAS/UAC cannot detect the original Call-ID, nor whether it has been changed.

There is no known security issue with viewing or modifying the Session-ID, other than to hamper Troubleshooting efforts.

9.2. Security considerations for extensions to the Session-ID

In general, B2BUA behavior cannot be dictated by standards. They do whatever their owners/operators wish them to do, or whatever is necessary to make their applications work. This document attempts to normatively specify B2BUA behavior, by creating a SIP header value for which the properties are such that B2BUA’s should have no legitimate reason to interfere. This effectively creates a "promise" that future uses of this Session-ID header field, including its value *and* future defined parameters, maintain this benign property. Any future extensions to the Session-ID mechanism and header field MUST maintain this property, or else B2BUA’s will begin to modify it again or remove it, and its value will be lost.

Manufacturers of SIP devices should note that there is no guarantee that a B2BUA will not inspect the Session-ID header field, and remove it if it does not comply with this document’s value restrictions. Because of this, any uses for Session-ID header parameters MUST be documented in RFCs.
10. IANA Considerations

This document asks IANA for a new SIP header field, in long and compact form.

11. Acknowledgments

Thanks to Raphael C EFFIC, Bob Penfield, Dale Worley, Paul Kyzivat, and Ian Elz for their input.

12. References

12.1. Normative References


12.2. Informative References


Author’s Address

Hadriel Kaplan
Acme Packet
71 Third Ave.
Burlington, MA 01803, USA

Email: hkaplan@acmepacket.com