

sip
Internet Draft
Document: [draft-willis-sip-cookies-00.txt](#)
Category: Standards Track

D. Willis
dynamicSoft
B. Rosen
Marconi
July, 2001

SIP Cookies

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#) [1].

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/1id-abstracts.txt>

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

1. Abstract

This document describes an extension to SIP ([RFC 2543](#)) to provide a mechanism for storing state and other information in SIP comparable to the HTTP state mechanisms defined in [RFC 2109](#). This extension includes a new SIP header ("Cookie:"), an option tag for feature negotiation ("cookie") and IANA registration considerations for registering the "cookie" extension, semantics of the value of the Cookie: header, and behavioral rules for processing these headers by SIP nodes supporting the "cookie" extension. This document also briefly discusses possible uses of cookies, and security considerations for their use.

2. Conventions used in this document

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](#) [2].

[3](#). Introduction

Willis	Standards Track	Expires February 2002	1
		SIP Cookies	July 2001

This is a standards-track document defining an extension to SIP ([RFC 2543](#)) for transmission of session state information.

SIP was initially derived from HTTP ([RFC 2616](#)) and shares much of its design rational. [RFC 2109](#) defines a mechanism (HTTP cookies) for transmitting session state information over HTTP and storing in endpoints for later use by servers. This document defines a similar mechanism for SIP, with similar rationale.

The cookie mechanism is appropriate for uses where a non-terminal SIP node, (such as a proxy) needs to insert application-specific information (called "state") into a session, which is transiting that node. This information may be retrieved (and potentially acted on) by any other node processing that message or further messages within the same SIP call, provided that the processing node is capable of comprehending the state information contained in the cookie. Cookies may be signed and/or encrypted as appropriate for applications on an application-specific basis.

[RFC 2543 section 10](#) defines SIP headers. Headers are essentially name-attribute string pairs that have three interesting properties for consideration in representing state information. [RFC 2543](#) distinguishes between "general header", which have defined broadly semantic properties in requests or responses, and "entity headers", which do not. It further defines a local extension policy, whereby a set of nodes may treat an entity header as a "general header" provided that they have agreement on its semantics.

Property 1, Extensibility: [RFC 2543](#) defines a set of standard header names and an IANA process for defining new option tags and associated headers. Implementors are free to add private or experimental headers conforming to the syntax referenced in [RFC 2543](#).

Property 2, Repeatability: There is no uniqueness requirement for a given header type within a single SIP message. That is, there may be several headers of the same type in a given message.

Property 3, Transparency: A SIP node that does not understand a particular header type is required to treat it as an "entity header". In general, proxies are required to transparently copy

entity headers during proxy operations, and endpoints simply ignore them if they do not understand that header.

Applications may in some cases insert transient state simply by encoding it into an extension header (an entity header with a locally unique name), such that an application node inserts the header into a SIP message as it passes through that node. Cookies differ from simple SIP extension headers in that they are more persistent. Unless renewed by a terminal node, extension headers exist only in the specific message into which they were inserted by

Willis

Standards Track û Expires February 2002

2

SIP Cookies

July 2001

an application, and no current mechanism exists for providing the information encoded into the header in further messages unless the terminal node has application awareness for that specific header. This is appropriate for many, but not all applications. Cookies provide for the preservation of state across the duration of a call and derivatives of that call. Rather than being copied into new messages relating to a call only when there is application awareness of the specific cookie (as with an extension header), cookies are copied into new messages within a call or derived from a call UNLESS there is specific application awareness that dictates that they not be.

When a terminal node (SIP User Agent) receives a Cookie: header, it stores the value of the cookie in an association with the call with which the message containing the cookie was associated. The terminal node then copies that cookie into every message it originates for the duration of that call or new call derived from that call (as in a redirect or transfer).

[4.](#) The "Cookie:" Header Syntax

The syntax for the Cookie: header is derived from the Set-Cookie syntax in [RFC 2109](#)

av-pairs	=	av-pair *("; " av-pair)
av-pair	=	attr ["=" value] ; optional value
attr	=	token
value	=	word
word	=	token quoted-string
cookie	=	"Cookie:" cookies
cookies	=	1#cookie
cookie	=	NAME "=" VALUE *("; " cookie-av)
NAME	=	attr
VALUE	=	value

```

cookie-av      =      "Comment" "=" value
                  |
                  "Domain" "=" value
                  |
                  "Version" "=" 1*DIGIT

```

Informally, the Cookie header comprises the token Cookie:, followed by a comma-separated list of one or more cookies. Each cookie begins with a NAME=VALUE pair, followed by zero or more semi-colon-separated attribute-value pairs. The specific attributes and the semantics of their values are defined below. The NAME=VALUE attribute-value pair must come first in each cookie. The others, if present, can occur in any order. If an attribute appears more than once in a cookie, the behavior is undefined.

NAME=VALUE

Required. The name of the state information ("cookie") is NAME, and its value is VALUE. NAMES that begin with \$ are reserved for other uses and must not be used by applications.

Willis	Standards Track û Expires February 2002	3
	SIP Cookies	July 2001

The VALUE may be opaque to the receiving node and may be anything the origin server chooses to send, possibly in a server-selected printable ASCII encoding. "Opaque" implies that the content is of interest and relevance only to the origin server or other nodes participating in the application. The content may, in fact, be readable by anyone that examines the Cookie header, but may have no specific meaning to them.

Comment=comment

Optional. Because cookies can contain private information about a user, the Cookie attribute allows an origin server to document its intended use of a cookie. The user can inspect the information to decide whether to initiate or continue a session with this cookie.

Domain=domain

Optional. The Domain attribute specifies the domain for which the cookie is valid (the originating domain). An explicitly specified domain must always start with a dot.

Version=version

Required. The Version attribute, a decimal integer, identifies to which version of the state management specification the cookie conforms. For this specification, Version=1 applies.

5. Behavior of SIP nodes receiving a Cookie

In general, a SIP node processing a message containing a cookie may modify or delete the cookie only if the node is participant in the application using the cookie and has adequate knowledge of the semantics of that specific cookie. We define such nodes as "participatory", and nodes without this involvement as "non-participatory". The processing of a cookie by a participatory node is subject to the requirements of the application using the cookie, and is therefore implementation dependent. Behavior for non-participatory SIP nodes is defined separately for proxies, user agents, and redirect servers.

Cookies are discarded when the call instantiating the cookie and all calls derived from that call have terminated..

5.1. Behavior of a non-participatory SIP Proxy Server receiving a Cookie

A SIP Proxy Server receiving a message containing a cookie pertaining to an application that is not relevant to this proxy treats the cookie as an unknown entity header according to the rules of [RFC 2543](#). In general, this means that the cookie is copied into any proxied message resulting from the incoming message.

Willis	Standards Track	Expires February 2002	4
		SIP Cookies	July 2001

5.2. Behavior of a non-participatory SIP redirect server receiving a Cookie

A SIP redirect server receiving a message containing a cookie not relevant to this redirect server must copy the cookie into any response, including redirection messages (300-class SIP messages) emitted as a result of the incoming message.

5.3 Behavior of a non-participatory User Agent receiving a Cookie

A SIP User Agent Server or User Agent Client receiving a message containing a cookie not relevant to this UA must store the cookie and include it any future messages emitted by the UA in the course of this call or derived calls. Derived calls here means any calls resulting from a redirection, re-invitation, referral (transfer) or any similar mechanism of the call with which the cookie was associated. If the received cookie differs in only the "Value" parameter from a cookie previously stored for this call, the UA replaces the stored cookie with the new cookie.

An example with UAs A and B and C and proxy P.

A invites B through P.
P attaches cookie K to the invitation and proxies it to B.
B stores the cookie and responds OK, including K in the response.
P proxies the OK to A, including K
A stores K.
A sends ACK to B, including K.
B sends BYE to A including K.
A sends OK to B including K.

Cookies are discarded when the call instantiating the cookie and all calls derived from that call have terminated..

[5.4](#) Implementation Limits

Practical User Agent implementations have limits on the number and size of cookies that they can store. In general, User Agents' cookie support should have no fixed limits. They should strive to store as many cookies as possible. Furthermore, general-use User Agents should provide each of the following minimum capabilities individually, although not necessarily simultaneously:

- * at least 10 cookies

- * at least 4096 bytes per cookie (as measured by the size of the characters that comprise the cookie non-terminal in the syntax description of the Cookie header)

- * at least 2 cookies per unique host or domain name

User agents created for specific purposes or for limited-capacity devices should provide at least 10 cookies of 4096 bytes.

Willis	Standards Track û Expires February 2002	5
	SIP Cookies	July 2001

The information in a Cookie must be retained in its entirety. If for some reason there is inadequate space to store the cookie, it must be discarded, not truncated.

Applications should use as few and as small cookies as possible, and they should cope gracefully with the loss of a cookie.

[5.4.1](#) Denial of Service Attacks

User Agents may choose to set an upper bound on the number of cookies to be stored from a given host or domain name or on the size of the cookie information. Otherwise a malicious node could attempt to flood a User Agent with many cookies, or large cookies, on successive responses, which would force out cookies the User Agent

had received from other servers. However, the minima specified above should still be supported.

6. Possible Usages of Cookie:

Cookies can be used for many applications requiring persistence of state preservation over a duration up to the lifetime of a call and its derived calls. Some applications will require participation only from the node originating the cookie, and others. Such uses might include:

- * State preservation in a call as proposed by the DCS "State" draft.
- * Associating a Billing-ID with a call as proposed by DCS Billing-ID Draft.
- * Tracking the changes in the target of a call (redirections and proxy operations) as proposed in "cc-redirect" draft.
- * Network-authenticated calling or called party identification as proposed by DCS Privacy draft.
- * Media authorization tokens as proposed by DCS "Call Auth" draft.

7. Option Tag for Cookie and IANA Considerations

[RFC 2543](#) establishes the IANA considerations for definition of a new SIP option tag. This option tag is used in server features negotiation (the Requires and Supports headers).

7.1. Name and Description of Option:

Option tag = "cookie"

Description: This SIP option indicates the cookie extension mechanism as described in this document.

7.2. New SIP Headers:

This extension adds the SIP header "Cookie:". The syntax of the Cookie: header is defined elsewhere in this document.

7.3. Change Control

The SIP Working Group of the IETF retains change control over the cookie extension to SIP.

7.4. Further Description of Extension

This document provides the detailed description and definition of the SIP cookie extension.

8. Security Considerations

The body of a SIP message may be read by any node participating in the session, and in the absence of transport-layer protection, by any intermediary on the IP network. Consequently, special attention must be applied to preserve the integrity or confidentiality of cookie names and values as appropriate to the information therein. It is suggested that implementations apply encryption using public or shared secret key techniques to sensitive information. Furthermore, nodes cannot be trusted not to alter the value of a cookie or insert falsely attributed cookies, and it may therefore be necessary to include a signing mechanism such as SSA or SHA/5 to the cookie. There may be further considerations for protection of messages at the SIP security level.

Cookies in SIP, unlike cookies in HTTP, are discarded at the conclusion of a session. Therefore, many of the privacy concerns of HTTP cookies do not apply to SIP cookies. However, SIP cookies could be used to track user activity throughout a session, which some users may consider to be a privacy concern. Some of the controls listed in [RFC2109](#) may therefore be appropriate.

9. Open Questions

9.1. Can a participatory node expecting a cookie reject a message which does not have the cookie or has a cookie with an inappropriate value? If so, how is this indicated?

10. References

- 1 Handley, et. al., "Session Initiation Protocol", [RFC 2543](#), March 1999.
- 2 Fielding, et. al., "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), June 1999.
- 3 Kristol, D., Montulli, L., "HTTP State Management Mechanism", [RFC 2109](#), February 1997

11. Acknowledgments

12. Author's Addresses

Dean Willis
dynamicSoft
<Company Address>
Phone: <optional>
Email: dwillis@dynamicsoft.com

Brian Rosen
Marconi
1000 Marconi Drive
Warrendale, PA 15096
USA
Phone: +1 724 742 6826
Email: brian.rosen@marconi.com

Full Copyright Statement

"Copyright (C) The Internet Society (date). All Rights Reserved.
This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into

