Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of 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/1id-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 May 14, 2008.

Copyright Notice

Copyright (C) The IETF Trust (2007).
Internet-Draft                  HTTP/1.1                   November 2007

Abstract

The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, collaborative, hypermedia information systems. HTTP has been in use by the World Wide Web global information initiative since 1990. This document is Part 7 of the eight-part specification that defines the protocol referred to as "HTTP/1.1" and, taken together, updates RFC 2616 and RFC 2617. Part 7 defines HTTP Authentication.

Table of Contents

1. Introduction ............................................. 3
2. Header Field Definitions ................................. 3
   2.1. Authorization ....................................... 3
   2.2. Proxy-Authenticate ................................. 4
   2.3. Proxy-Authorization ............................... 4
   2.4. WWW-Authenticate ................................. 5
3. IANA Considerations ...................................... 5
4. Security Considerations ................................. 5
   4.1. Authentication Credentials and Idle Clients ........ 5
5. Acknowledgments ......................................... 6
6. References ............................................... 6
Index ....................................................... 6
Authors' Addresses ........................................... 7
Intellectual Property and Copyright Statements ........... 9
1. Introduction

This document will define aspects of HTTP related to access control and authentication. Right now it only includes the extracted relevant sections of RFC 2616 [RFC2616] with only minor edits.

HTTP provides several OPTIONAL challenge-response authentication mechanisms which can be used by a server to challenge a client request and by a client to provide authentication information. The general framework for access authentication, and the specification of "basic" and "digest" authentication, are specified in "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617]. This specification adopts the definitions of "challenge" and "credentials" from that specification.

2. Header Field Definitions

This section defines the syntax and semantics of all standard HTTP/1.1 header fields. For entity-header fields, both sender and recipient refer to either the client or the server, depending on who sends and who receives the entity.

2.1. Authorization

A user agent that wishes to authenticate itself with a server--usually, but not necessarily, after receiving a 401 response--does so by including an Authorization request-header field with the request. The Authorization field value consists of credentials containing the authentication information of the user agent for the realm of the resource being requested.

Authorization = "Authorization" "::" credentials

HTTP access authentication is described in "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617]. If a request is authenticated and a realm specified, the same credentials SHOULD be valid for all other requests within this realm (assuming that the authentication scheme itself does not require otherwise, such as credentials that vary according to a challenge value or using synchronized clocks).

When a shared cache (see [Part 6]) receives a request containing an Authorization field, it MUST NOT return the corresponding response as a reply to any other request, unless one of the following specific exceptions holds:
1. If the response includes the "s-maxage" cache-control directive, the cache MAY use that response in replying to a subsequent request. But (if the specified maximum age has passed) a proxy cache MUST first revalidate it with the origin server, using the request-headers from the new request to allow the origin server to authenticate the new request. (This is the defined behavior for s-maxage.) If the response includes "s-maxage=0", the proxy MUST always revalidate it before re-using it.

2. If the response includes the "must-revalidate" cache-control directive, the cache MAY use that response in replying to a subsequent request. But if the response is stale, all caches MUST first revalidate it with the origin server, using the request-headers from the new request to allow the origin server to authenticate the new request.

3. If the response includes the "public" cache-control directive, it MAY be returned in reply to any subsequent request.

2.2. Proxy-Authenticate

The Proxy-Authenticate response-header field MUST be included as part of a 407 (Proxy Authentication Required) response. The field value consists of a challenge that indicates the authentication scheme and parameters applicable to the proxy for this Request-URI.

   Proxy-Authenticate  = "Proxy-Authenticate" ":" 1#challenge

The HTTP access authentication process is described in "HTTP Authentication: Basic and Digest Access Authentication" [RFC2617]. Unlike WWW-Authenticate, the Proxy-Authenticate header field applies only to the current connection and SHOULD NOT be passed on to downstream clients. However, an intermediate proxy might need to obtain its own credentials by requesting them from the downstream client, which in some circumstances will appear as if the proxy is forwarding the Proxy-Authenticate header field.

2.3. Proxy-Authorization

The Proxy-Authorization request-header field allows the client to identify itself (or its user) to a proxy which requires authentication. The Proxy-Authorization field value consists of credentials containing the authentication information of the user agent for the proxy and/or realm of the resource being requested.

   Proxy-Authorization     = "Proxy-Authorization" ":" credentials

The HTTP access authentication process is described in "HTTP
Authentication: Basic and Digest Access Authentication" [RFC2617].
Unlike Authorization, the Proxy-Authorization header field applies
only to the next outbound proxy that demanded authentication using
the Proxy-Authenticate field. When multiple proxies are used in a
chain, the Proxy-Authorization header field is consumed by the first
outbound proxy that was expecting to receive credentials. A proxy
MAY relay the credentials from the client request to the next proxy
if that is the mechanism by which the proxies cooperatively
authenticate a given request.

2.4. WWW-Authenticate

The WWW-Authenticate response-header field MUST be included in 401
(Unauthorized) response messages. The field value consists of at
least one challenge that indicates the authentication scheme(s) and
parameters applicable to the Request-URI.

WWW-Authenticate  = "WWW-Authenticate" ";" 1#challenge

The HTTP access authentication process is described in "HTTP
Authentication: Basic and Digest Access Authentication" [RFC2617].
User agents are advised to take special care in parsing the WWW-
Authenticate field value as it might contain more than one challenge,
or if more than one WWW-Authenticate header field is provided, the
contents of a challenge itself can contain a comma-separated list of
authentication parameters.

3. IANA Considerations

TBD.

4. Security Considerations

This section is meant to inform application developers, information
providers, and users of the security limitations in HTTP/1.1 as
described by this document. The discussion does not include
definitive solutions to the problems revealed, though it does make
some suggestions for reducing security risks.

4.1. Authentication Credentials and Idle Clients

Existing HTTP clients and user agents typically retain authentication
information indefinitely. HTTP/1.1 does not provide a method for a
server to direct clients to discard these cached credentials. This
is a significant defect that requires further extensions to HTTP.
Circumstances under which credential caching can interfere with the
application's security model include but are not limited to:

- Clients which have been idle for an extended period following which the server might wish to cause the client to reprompt the user for credentials.

- Applications which include a session termination indication (such as a 'logout' or 'commit' button on a page) after which the server side of the application 'knows' that there is no further reason for the client to retain the credentials.

This is currently under separate study. There are a number of workarounds to parts of this problem, and we encourage the use of password protection in screen savers, idle time-outs, and other methods which mitigate the security problems inherent in this problem. In particular, user agents which cache credentials are encouraged to provide a readily accessible mechanism for discarding cached credentials under user control.

5. Acknowledgments

Based on an XML translation of [RFC2616] by Julian Reschke.

6. References


Index

A
  Authorization header  3

G
  Grammar
  Authorization  3
  Proxy-Authenticate  4
  Proxy-Authorization  4
  WWW-Authenticate  5
H

Headers
  Authorization  3
  Proxy-Authenticate  4
  Proxy-Authorization  4
  WWW-Authenticate  5

P

Proxy-Authenticate header  4
Proxy-Authorization header  4

W

WWW-Authenticate header  5

Authors' Addresses

Roy T. Fielding (editor)
Day Software
23 Corporate Plaza DR, Suite 280
Newport Beach, CA  92660
USA

Phone: +1-949-706-5300
Fax:   +1-949-706-5305
Email: fielding@gbiv.com
URI:   http://roy.gbiv.com/

James Gettys
Hewlett-Packard Company
HP Labs, Cambridge Research Laboratory
One Cambridge Center
Cambridge, MA  02138
USA

Email: Jim.Gettys@hp.com

Jeffrey C. Mogul
Hewlett-Packard Company
HP Labs, Large Scale Systems Group
1501 Page Mill Road, MS 1177
Palo Alto, CA  94304
USA

Email: JeffMogul@acm.org
Henrik Frystyk Nielsen
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052
USA

Email: henrikn@microsoft.com

Larry Masinter
Adobe Systems, Incorporated
345 Park Ave
San Jose, CA 95110
USA

Email: LMM@acm.org
URI: http://larry.masinter.net/

Paul J. Leach
Microsoft Corporation
1 Microsoft Way
Redmond, WA 98052

Email: paulle@microsoft.com

Tim Berners-Lee
World Wide Web Consortium
MIT Laboratory for Computer Science
545 Technology Square
Cambridge, MA 02139
USA

Fax: +1 (617) 258 8682
Email: timbl@w3.org