HTTPbis Working Group
Internet-Draft
Obsoletes: 2616 (if approved)
Updates: 2617 (if approved)
Intended status: Standards Track
Expires: April 28, 2011

R. Fielding, Ed.
Day Software
J. Gettys
Alcatel-Lucent
J. Mogul
HP
H. Frystyk
Microsoft
L. Masinter
Adobe Systems
P. Leach
Microsoft
T. Berners-Lee
W3C/MIT
Y. Lafon, Ed.
W3C
J. Reschke, Ed.
greenbytes
October 25, 2010

HTTP/1.1, part 7: Authentication
draft-ietf-httpbis-p7-auth-12

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 seven-part specification that defines the protocol referred to as "HTTP/1.1" and, taken together, obsoletes RFC 2616. Part 7 defines HTTP Authentication.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft should take place on the HTTPBIS working group mailing list (ietf-http-wg@w3.org). The current issues list is at <http://tools.ietf.org/wg/httpbis/trac/report/3> and related documents (including fancy diffs) can be found at <http://tools.ietf.org/wg/httpbis/>.

The changes in this draft are summarized in Appendix B.13.

Status of This Memo

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

Copyright Notice

Table of Contents

1. Introduction

This document defines HTTP/1.1 access control and authentication. It includes the relevant parts of RFC 2616 with only minor changes, plus the general framework for HTTP authentication, as previously defined in "HTTP Authentication: Basic and Digest Access Authentication" ([RFC2617]).

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 "basic" and "digest" authentication schemes continue to be specified in RFC 2617.

1.1. Requirements

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].

An implementation is not compliant if it fails to satisfy one or more of the "MUST" or "REQUIRED" level requirements for the protocols it implements. An implementation that satisfies all the "MUST" or "REQUIRED" level and all the "SHOULD" level requirements for its protocols is said to be "unconditionally compliant"; one that satisfies all the "MUST" level requirements but not all the "SHOULD" level requirements for its protocols is said to be "conditionally compliant".

1.2. Syntax Notation

This specification uses the ABNF syntax defined in Section 1.2 of [Part1] (which extends the syntax defined in [RFC5234] with a list rule). Appendix A shows the collected ABNF, with the list rule expanded.

The following core rules are included by reference, as defined in [RFC5234], Appendix B.1: ALPHA (letters), CR (carriage return), CRLF (CR LF), CTL (controls), DIGIT (decimal 0-9), DQUOTE (double quote), HEXDIG (hexadecimal 0-9/A-F/a-f), LF (line feed), OCTET (any 8-bit sequence of data), SP (space), VCHAR (any visible USASCII character), and WSP (whitespace).

1.2.1. Core Rules

The core rules below are defined in Section 1.2.2 of [Part1]:


```
quoted-string = <quoted-string, defined in [Part1], Section 1.2.2>
token         = <token, defined in [Part1], Section 1.2.2>
OWS           = <OWS, defined in [Part1], Section 1.2.2>
```

2. Access Authentication Framework

HTTP provides a simple challenge-response authentication mechanism that can be used by a server to challenge a client request and by a client to provide authentication information. It uses an extensible, case-insensitive token to identify the authentication scheme, followed by a comma-separated list of attribute-value pairs which carry the parameters necessary for achieving authentication via that scheme.

```
auth-scheme    = token
auth-param     = token "=" ( token / quoted-string )
```

The 401 (Unauthorized) response message is used by an origin server to challenge the authorization of a user agent. This response MUST include a WWW-Authenticate header field containing at least one challenge applicable to the requested resource. The 407 (Proxy Authentication Required) response message is used by a proxy to

challenge the authorization of a client and MUST include a Proxy-Authenticate header field containing at least one challenge applicable to the proxy for the requested resource.

```
  challenge   = auth-scheme 1*SP 1#auth-param
```

Note: User agents will need to take special care in parsing the WWW-Authenticate or Proxy-Authenticate header field value if it contains more than one challenge, or if more than one WWW-Authenticate header field is provided, since the contents of a challenge can itself contain a comma-separated list of authentication parameters.

Note: Many browsers fail to parse challenges containing unknown schemes. A workaround for this problem is to list well-supported schemes (such as "basic") first.

The authentication parameter realm is defined for all authentication schemes:

```
  realm       = "realm" "=" realm-value
  realm-value = quoted-string
```

The realm directive (case-insensitive) is required for all authentication schemes that issue a challenge. The realm value (case-sensitive), in combination with the canonical root URI (the

scheme and authority components of the effective request URI; see Section 4.3 of [Part1]) of the server being accessed, defines the protection space. These realms allow the protected resources on a server to be partitioned into a set of protection spaces, each with its own authentication scheme and/or authorization database. The realm value is a string, generally assigned by the origin server, which can have additional semantics specific to the authentication scheme. Note that there can be multiple challenges with the same auth-scheme but different realms.

A user agent that wishes to authenticate itself with an origin server -- usually, but not necessarily, after receiving a 401 (Unauthorized) -- MAY do so by including an Authorization header field with the request. A client that wishes to authenticate itself with a proxy -- usually, but not necessarily, after receiving a 407 (Proxy

Authentication Required) -- MAY do so by including a Proxy-Authorization header field with the request. Both the Authorization field value and the Proxy-Authorization field value consist of credentials containing the authentication information of the client for the realm of the resource being requested. The user agent MUST choose to use one of the challenges with the strongest auth-scheme it understands and request credentials from the user based upon that challenge.

```
credentials = auth-scheme ( token
                          / quoted-string
                          / #auth-param )
```

The protection space determines the domain over which credentials can be automatically applied. If a prior request has been authorized, the same credentials MAY be reused for all other requests within that protection space for a period of time determined by the authentication scheme, parameters, and/or user preference. Unless otherwise defined by the authentication scheme, a single protection space cannot extend outside the scope of its server.

If the origin server does not wish to accept the credentials sent with a request, it SHOULD return a 401 (Unauthorized) response. The response MUST include a WWW-Authenticate header field containing at least one (possibly new) challenge applicable to the requested resource. If a proxy does not accept the credentials sent with a request, it SHOULD return a 407 (Proxy Authentication Required). The response MUST include a Proxy-Authenticate header field containing a (possibly new) challenge applicable to the proxy for the requested resource.

The HTTP protocol does not restrict applications to this simple challenge-response mechanism for access authentication. Additional

mechanisms MAY be used, such as encryption at the transport level or via message encapsulation, and with additional header fields specifying authentication information. However, these additional mechanisms are not defined by this specification.

Proxies MUST be completely transparent regarding user agent authentication by origin servers. That is, they MUST forward the WWW-Authenticate and Authorization headers untouched, and follow the

rules found in Section 4.1. Both the Proxy-Authenticate and the Proxy-Authorization header fields are hop-by-hop headers (see Section 7.1.3.1 of [Part1]).

2.1. Authentication Scheme Registry

The HTTP Authentication Scheme Registry defines the name space for the authentication schemes in challenges and credentials.

Registrations MUST include the following fields:

- Authentication Scheme Name
- Pointer to specification text

Values to be added to this name space are subject to IETF review ([RFC5226], Section 4.1).

The registry itself is maintained at <http://www.iana.org/assignments/http-authschemes>.

3. Status Code Definitions

3.1. 401 Unauthorized

The request requires user authentication. The response MUST include a WWW-Authenticate header field (Section 4.4) containing a challenge applicable to the target resource. The client MAY repeat the request with a suitable Authorization header field (Section 4.1). If the request already included Authorization credentials, then the 401 response indicates that authorization has been refused for those credentials. If the 401 response contains the same challenge as the prior response, and the user agent has already attempted authentication at least once, then the user SHOULD be presented the representation that was given in the response, since that representation might include relevant diagnostic information.

3.2. 407 Proxy Authentication Required

This code is similar to 401 (Unauthorized), but indicates that the client ought to first authenticate itself with the proxy. The proxy MUST return a Proxy-Authenticate header field (Section 4.2) containing a challenge applicable to the proxy for the target resource. The client MAY repeat the request with a suitable Proxy-Authorization header field (Section 4.3).

4. Header Field Definitions

This section defines the syntax and semantics of HTTP/1.1 header fields related to authentication.

4.1. Authorization

The "Authorization" request-header field allows a user agent to authenticate itself with a server -- usually, but not necessarily, after receiving a 401 (Unauthorized) response. Its value consists of credentials containing information of the user agent for the realm of the resource being requested.

```
  Authorization   = "Authorization" ":" OWS Authorization-v
  Authorization-v = credentials
```

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 Section 1.2 of [Part6]) 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-header fields 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-header fields 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.

4.2. Proxy-Authenticate

The "Proxy-Authenticate" response-header field consists of a challenge that indicates the authentication scheme and parameters applicable to the proxy for this effective request URI (Section 4.3 of [Part1]). It MUST be included as part of a 407 (Proxy Authentication Required) response.

```
  Proxy-Authenticate   = "Proxy-Authenticate" ":" OWS
                         Proxy-Authenticate-v
  Proxy-Authenticate-v = 1#challenge
```

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.

4.3. Proxy-Authorization

The "Proxy-Authorization" request-header field allows the client to identify itself (or its user) to a proxy which requires authentication. Its 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" ":" OWS
                          Proxy-Authorization-v
  Proxy-Authorization-v = credentials
```

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.

4.4. WWW-Authenticate

The "WWW-Authenticate" response-header field consists of at least one challenge that indicates the authentication scheme(s) and parameters applicable to the effective request URI (Section 4.3 of [Part1]). It MUST be included in 401 (Unauthorized) response messages.

```
  WWW-Authenticate   = "WWW-Authenticate" ":" OWS WWW-Authenticate-v
  WWW-Authenticate-v = 1#challenge
```

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.

5. IANA Considerations

5.1. Authenticaton Scheme Registry

The registration procedure for HTTP Authentication Schemes is defined by Section 2.1 of this document.

The HTTP Method Authentication Scheme shall be created at <http://www.iana.org/assignments/http-authschemes>.

5.2. Status Code Registration

The HTTP Status Code Registry located at <http://www.iana.org/assignments/http-status-codes> shall be updated with the registrations below:

Value	Description	Reference
401	Unauthorized	Section 3.1
407	Proxy Authentication Required	Section 3.2

5.3. Header Field Registration

The Message Header Field Registry located at <http://www.iana.org/assignments/message-headers/message-header-index.html> shall be updated with the permanent registrations below (see [RFC3864]):

Header Field Name	Protocol	Status	Reference
Authorization	http	standard	Section 4.1
Proxy-Authenticate	http	standard	Section 4.2
Proxy-Authorization	http	standard	Section 4.3
WWW-Authenticate	http	standard	Section 4.4

The change controller is: "IETF (iesg@ietf.org) - Internet Engineering Task Force".

6. 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.

6.1. 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 work-arounds 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.

7. Acknowledgments

This specification takes over the definition of the HTTP Authentication Framework, previously defined in RFC 2617. We thank to John Franks, Phillip M. Hallam-Baker, Jeffery L. Hostetler, Scott D. Lawrence, Paul J. Leach, Ari Luotonen, and Lawrence C. Stewart for their work on that specification.

[[acks: HTTPbis acknowledgements.]]

8. References

8.1. Normative References

[Part1] Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., and J. Reschke, Ed., "HTTP/1.1, part 1: URIs, Connections, and Message Parsing", draft-ietf-httpbis-p1-messaging-12 (work in progress), October 2010.

[Part6] Fielding, R., Ed., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., Lafon, Y., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1, part 6: Caching", draft-ietf-httpbis-p6-cache-12 (work in progress), October 2010.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

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

8.2. Informative References

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999.

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., Lawrence, S., Leach, P., Luotonen, A., and L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999.

[RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an

IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

Appendix A. Collected ABNF

```
Authorization = "Authorization:" OWS Authorization-v
Authorization-v = credentials

OWS = <OWS, defined in [Part1], Section 1.2.2>

Proxy-Authenticate = "Proxy-Authenticate:" OWS Proxy-Authenticate-v
Proxy-Authenticate-v = *( "," OWS ) challenge *( OWS "," [ OWS
 challenge ] )
Proxy-Authorization = "Proxy-Authorization:" OWS
 Proxy-Authorization-v
Proxy-Authorization-v = credentials

WWW-Authenticate = "WWW-Authenticate:" OWS WWW-Authenticate-v
```

```
WWW-Authenticate-v = *( "," OWS ) challenge *( OWS "," [ OWS
 challenge ] )

auth-param = token "=" ( token / quoted-string )
auth-scheme = token

challenge = auth-scheme 1*SP *( "," OWS ) auth-param *( OWS "," [ OWS
 auth-param ] )
credentials = auth-scheme ( token / quoted-string / [ ( "," /
 auth-param ) *( OWS "," [ OWS auth-param ] ) ] )

quoted-string = <quoted-string, defined in [Part1], Section 1.2.2>

realm = "realm=" realm-value
realm-value = quoted-string

token = <token, defined in [Part1], Section 1.2.2>

ABNF diagnostics:

; Authorization defined but not used
; Proxy-Authenticate defined but not used
; Proxy-Authorization defined but not used
; WWW-Authenticate defined but not used
; realm defined but not used
```


Appendix B. Change Log (to be removed by RFC Editor before publication)

B.1. Since RFC 2616

Extracted relevant partitions from [RFC2616].

B.2. Since draft-ietf-httpbis-p7-auth-00

Closed issues:

- <http://tools.ietf.org/wg/httpbis/trac/ticket/35>: "Normative and

Informative references"

B.3. Since draft-ietf-httpbis-p7-auth-01

Ongoing work on ABNF conversion (<http://tools.ietf.org/wg/httpbis/trac/ticket/36>):

- Explicitly import BNF rules for "challenge" and "credentials" from RFC2617.

- Add explicit references to BNF syntax and rules imported from other parts of the specification.

B.4. Since draft-ietf-httpbis-p7-auth-02

Ongoing work on IANA Message Header Field Registration (<http://tools.ietf.org/wg/httpbis/trac/ticket/40>):

- Reference RFC 3984, and update header field registrations for header fields defined in this document.

B.5. Since draft-ietf-httpbis-p7-auth-03

B.6. Since draft-ietf-httpbis-p7-auth-04

Ongoing work on ABNF conversion (<http://tools.ietf.org/wg/httpbis/trac/ticket/36>):

- Use "/" instead of "|" for alternatives.

- Introduce new ABNF rules for "bad" whitespace ("BWS"), optional whitespace ("OWS") and required whitespace ("RWS").

- Rewrite ABNFs to spell out whitespace rules, factor out header field value format definitions.

B.7. Since draft-ietf-httpbis-p7-auth-05

Final work on ABNF conversion (<http://tools.ietf.org/wg/httpbis/trac/ticket/36>):

- Add appendix containing collected and expanded ABNF, reorganize ABNF introduction.

B.8. Since draft-ietf-httpbis-p7-auth-06

None.

B.9. Since draft-ietf-httpbis-p7-auth-07

Closed issues:

- <http://tools.ietf.org/wg/httpbis/trac/ticket/198>: "move IANA registrations for optional status codes"

B.10. Since draft-ietf-httpbis-p7-auth-08

No significant changes.

B.11. Since draft-ietf-httpbis-p7-auth-09

Partly resolved issues:

- <http://tools.ietf.org/wg/httpbis/trac/ticket/196>: "Term for the requested resource's URI"

B.12. Since draft-ietf-httpbis-p7-auth-10

None yet.

B.13. Since draft-ietf-httpbis-p7-auth-11

Closed issues:

- <http://tools.ietf.org/wg/httpbis/trac/ticket/130>: "introduction to part 7 is work-in-progress"
- <http://tools.ietf.org/wg/httpbis/trac/ticket/195>: "auth-param syntax"
- <http://tools.ietf.org/wg/httpbis/trac/ticket/237>: "absorbing the auth framework from 2617"

Partly resolved issues:

- <http://tools.ietf.org/wg/httpbis/trac/ticket/141>: "should we have an auth scheme registry"

Index

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/

Jim Gettys
Alcatel-Lucent Bell Labs
21 Oak Knoll Road
Carlisle, MA 01741
USA

EMail: jg@freedesktop.org
URI: http://gettys.wordpress.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 Computer Science and Artificial Intelligence Laboratory
The Stata Center, Building 32
32 Vassar Street
Cambridge, MA 02139
USA

EMail: timbl@w3.org
URI: http://www.w3.org/People/Berners-Lee/

Yves Lafon (editor)
World Wide Web Consortium
W3C / ERCIM

2004, rte des Lucioles
Sophia-Antipolis, AM 06902
France

EMail: ylafon@w3.org
URI: http://www.raubacapeu.net/people/yves/

Julian F. Reschke (editor)
greenbytes GmbH
Hafenweg 16
Muenster, NW 48155
Germany

Phone: +49 251 2807760
Fax: +49 251 2807761
EMail: julian.reschke@greenbytes.de
URI: http://greenbytes.de/tech/webdav/