

INTERNET-DRAFT
Intended Status: Standards Track
Expires: January 31, 2013

David Ahrens
Rifaat Shekh-Yusef
Avaya
July 30, 2012

**HTTP Digest Access Authentication Algorithm Update
draft-ahrens-httpbis-digest-auth-update-00**

Abstract

This document specifies extensions to the HTTP Digest Authentication mechanism to add support for the SHA1 and SHA2 digest algorithms to the HTTP Digest access authentication scheme.

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/lid-abstracts.html>

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

Copyright and License 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	Introduction	3
1.1	Terminology	3
2	Digest Access Authentication Scheme Update	3
2.1	Introduction	3
2.1.1	Representation of digest values	3
2.1.2	Limitations	4
2.2	Specification of Digest Headers	4
2.2.1	The WWW-Authenticate Response Header	4
2.2.2	The Authorization Request Header	5
2.3	Digest Operation	6
2.4	Security Protocol Operation	6
2.5	Example	6
3	Security Considerations	7
4	Acknowledgments	7
5	References	8
5.1	Normative References	8
5.2	Informative References	8
6	Authors' Addresses	9

1 Introduction

This document specifies extensions to the HTTP Digest Access Authentication scheme by adding support for the SHA1 and SHA2 suite of hash algorithms. [RFC 2617](#) specifies the MD5 algorithm as the default hash algorithm used in the digest access authentication scheme. Since [RFC 2617](#) was first proposed, the MD5 algorithm has been broken. In 2008 the US-CERT issued a note that MD5 "should be considered cryptographically broken and unsuitable for further use."

1.1 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](#) [[RFC2119](#)].

2. Digest Access Authentication Scheme Update

The Digest Access Authentication scheme is based on a simple challenge-response paradigm. The Digest scheme challenges using a nonce value. A valid response contains a checksum of the username, the password, the given nonce value and the requested URI. In this way the password is never sent in the clear.

2.1 Introduction

2.1.1 Representation of digest values

An optional header allows the server to specify the algorithm used to create the checksum or digest. By default and to maintain backwards compatibility, the MD5 algorithm is used.

The size of the digest depends on the algorithm used. The bits in the digest are converted from the most significant to the least significant bit, four bits at a time to the ASCII representation as follows. Each four bits is represented by its familiar hexadecimal notation from the characters 0123456789abcdef, that is binary 0000 is represented by the character '0', 0001 by '1' and so on up to the representation of 1111 as 'f'. If the MD5 algorithm is used to calculate the digest, then the digest will be represented as 32 hexadecimal characters, SHA1 by 40 hexadecimal characters, SHA2-224 by 56 hexadecimal characters, SHA2-256 by 64 hexadecimal characters, SHA2-384 by 96 hexadecimal characters, and SHA2-512 by 128 hexadecimal characters.

[2.1.2](#) Limitations

The Digest authentication scheme specified in [RFC 2617](#) suffers from many known limitations. The update proposed in this draft does not address those limitations.

[2.2](#) Specification of Digest Headers

The modifications to the formats of the WWW-Authenticate Header line and the Authorization header line are specified below.

[2.2.1](#) The WWW-Authenticate Response Header

If a server receives a request for an access protected object, and an acceptable Authorization header is not sent, the server responds with a "401 Unauthorized" status code, and a WWW-Authenticate header. The server MAY include multiple WWW-Authenticate headers to allow the server to utilize the best available algorithm supported by the client.

The algorithm directive is extended as follows:

```
algorithm = "algorithm" "=" ( "MD5" | "MD5-sess" |  
                                "SHA1" | "SHA1-sess" |  
                                "SHA224" | "SHA224-sess" |  
                                "SHA256" | "SHA256-sess" |  
                                "SHA384" | "SHA384-sess" |  
                                "SHA512" | "SHA512-sess" |  
                                token )
```

Algorithm

A string indicating a pair of algorithms used to produce the digest and a checksum. If the algorithm parameter is not present it is assumed to be "MD5" to maintain backwards compatibility with existing implementations. If the algorithm is not understood, the challenge should be ignored and a different challenge used if there is more than one.

The string obtained by applying the digest algorithm to the data "data" with "secret" will be denoted KD(secret, data) and the string obtained by applying the checksum algorithm to the data "data" will be denoted H(data). The notation unq(x) means the value of the quoted string X without the surrounding quotes.

For the "MD5 and "MD5-sess" algorithms
H(data) = MD5(data)

For the "SHA1" and "SHA1-sess" algorithms
H(data) = SHA1(data)

For the "SHA224" and "SHA224-sess" algorithms
H(data) = SHA224(data)

For the "SHA256" and "SHA256-sess" algorithms
H(data) = SHA256(data)

For the "SHA384" and "SHA384-sess" algorithms
H(data) = SHA384(data)

For the "SHA512" and "SHA512-sess" algorithms
H(data) = SHA512(data)

and

KD(secret, data) = H(concat(secret, ":", data))

i.e the digest is the hash of the secret concatenated with a colon concatenated with the data. The "-sess" algorithm is intended to allow efficient 3rd party authentication servers; for the difference in usage see the description in section [RFC2617, Section 3.2.2.2](#).

2.2.2 The Authorization Request Header

The client is expected to retry the request, passing an Authorization Request Header line. The Authorization Request Header line is modified as follows:

```
request-digest      = <"> digest-size LHEX <">
digest-size        = "32" | "40" | "56" | "64" | "96" | "128"
```

The values of the opaque and algorithm fields must match those supplied in the WWW-Authenticate response header for the entity being requested.

response

A string of hex digits as defined in [RFC2617](#) which proves that the user knows a password.

2.3 Digest Operation

The modifications specified in this document does not introduce any change to the digest operation specified in [RFC2617](#).

2.4 Security Protocol Operation

When a server receives a request to access a resource, the server might challenge the client by responding with "401 Unauthorized" status code, and include one or more WWW-Authenticate headers. If the server challenges with multiple Digest headers, then each one of these headers MUST use a different digest algorithm. The server MUST add these Digest headers to the response in order of preference, starting with the most preferred header, followed by the less preferred headers.

When the client receives the response it SHOULD use the topmost header that it supports, unless a local policy dictates otherwise. The client should ignore any challenge it does not understand.

2.5 Example

The following example is borrowed from [RFC 2617](#) and assumes that an access protected document is being requested from the server via a GET request. The URI of the document is <http://www.nowhere.org/dir/index.html>. Both client and server know that the username for this document is "Mufasa" and the password is "Circle of Life" (with one space between each of the three words).

The first time the client requests the document, no Authorization header is sent, so the server responds with:

```
HTTP/1.1 401 Unauthorized
WWW-Authenticate: Digest
  realm = "testrealm@host.com",
  qop="auth, auth-int",
  algorithm="SHA1",
  nonce=dcd98b7102dd2f0e8b11d0f600bfb0c093",
  opaque=5ccc069c403ebaf9f0171e9517f40e41"

WWW-Authenticate: Digest
  realm="testrealm@host.com",
  qop="auth, auth-int",
  algorithm="MD5",
  nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",
  opaque="5ccc069c403ebaf9f0171e9517f40ef41"
```


The client may prompt the user for their username and password, after which it will respond with a new request, including the following Authorization header:

```
Authorization:Digest username="Mufasa",  
realm="testrealm@host.com"  
nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093",  
uri="/dir/index.html",  
qop=auth,  
algorithm="MD5"  
nc=00000001,  
cnonce="0a4f113b",  
response="6629fae49393a05397450978507c4ef1",  
opaque="5ccc069c403ebaf9f0171e9517f40e41"
```

3 Security Considerations

This specification updates the Digest Access Authentication scheme specified in [RFC 2617](#) to add support for the SHA1 and SHA2 algorithm suites. Support for these additional hash algorithms does not alter the security properties of the Digest Access Authentication scheme.

4 Acknowledgments

The authors would like to thank Geoff Baskwill and Eric Cooper for their careful review and comments.

5 References

5.1 Normative References

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

[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.

5.2 Informative References

6 Authors' Addresses

David Ahrens
Avaya, Inc.
4655 Great America Parkway
Santa Clara, CA 95054

Phone: (408) 562-5502
EMail: davidahrens@avaya.com

Rifaat Shekh-Yusef
Avaya, Inc.
250 Sydney Street
Belleville, Ontario
Canada

Phone: +1-613-967-5267
Email: rifatyu@avaya.com