Network Working Group
Request for Comments: 4635
Category: Standards Track
D. Eastlake 3rd
Motorola Laboratories
August 2006

HMAC SHA TSIG Algorithm Identifiers

Status of This Memo

Copyright Notice

Abstract

Use of the Domain Name System TSIG resource record requires specification of a cryptographic message authentication code. Currently, identifiers have been specified only for HMAC MD5 (Hashed Message Authentication Code, Message Digest 5) and GSS (Generic Security Service) TSIG algorithms. This document standardizes identifiers and implementation requirements for additional HMAC SHA (Secure Hash Algorithm) TSIG algorithms and standardizes how to specify and handle the truncation of HMAC values in TSIG.

Table of Contents

1. Introduction

[RFC2845] specifies a TSIG Resource Record (RR) that can be used to authenticate DNS (Domain Name System [STD13]) queries and responses. This RR contains a domain name syntax data item that names the authentication algorithm used. [RFC2845] defines the HMAC-MD5.SIG-ALG.REG.INT name for authentication codes using the HMAC (Hashed Message Authentication Code) [RFC2104] algorithm with the MD5 (Message Digest 5) [RFC1321] hash algorithm. IANA has also registered "gss-tsig" as an identifier for TSIG authentication where the cryptographic operations are delegated to the Generic Security Service (GSS) [RFC3645].

Note that use of TSIG presumes prior agreement, between the resolver and server involved, as to the algorithm and key to be used.

In Section 2, this document specifies additional names for TSIG authentication algorithms based on US NIST SHA (United States, National Institute of Science and Technology, Secure Hash Algorithm) algorithms and HMAC and specifies the implementation requirements for those algorithms.

In Section 3, this document specifies the effect of inequality between the normal output size of the specified hash function and the length of MAC (Message Authentication Code) data given in the TSIG RR. In particular, it specifies that a shorter-length field value specifies truncation and that a longer-length field is an error.

In Section 4, policy restrictions and implications related to truncation are described and specified, as is a new error code to indicate truncation shorter than that permitted by policy.

The key words "MUST", "MUST NOT", "SHOULD", "SHOULD NOT", "MAY", in this document are to be interpreted as described in [RFC2119].

2. Algorithms and Identifiers

TSIG Resource Records (RRs) [RFC2845] are used to authenticate DNS queries and responses. They are intended to be efficient symmetric authentication codes based on a shared secret. (Asymmetric signatures can be provided using the SIG RR [RFC2931]. In particular, SIG(0) can be used for transaction signatures.) Used with a strong hash function, HMAC [RFC2104] provides a way to calculate such symmetric authentication codes. The only specified HMAC-based TSIG algorithm identifier has been HMAC-MD5.SIG-ALG.REG.INT, based on MD5 [RFC1321].

The use of SHA-1 [FIPS180-2, RFC3174], which is a 160-bit hash, as compared with the 128 bits for MD5, and additional hash algorithms in the SHA family [FIPS180-2, RFC3874, RFC4634] with 224, 256, 384, and 512 bits may be preferred in some cases. This is because increasingly successful cryptanalytic attacks are being made on the shorter hashes.

Use of TSIG between a DNS resolver and server is by mutual agreement. That agreement can include the support of additional algorithms and criteria as to which algorithms and truncations are acceptable, subject to the restriction and guidelines in Sections 3 and 4 below. Key agreement can be by the TKEY mechanism [RFC2930] or some other mutually agreeable method.

The current HMAC-MD5.SIG-ALG.REG.INT and gss-tsig identifiers are included in the table below for convenience. Implementations that support TSIG MUST also implement HMAC SHA1 and HMAC SHA256 and MAY implement gss-tsig and the other algorithms listed below.

Mandatory	HMAC-MD5.SIG-ALG.REG.INT
Optional	gss-tsig
Mandatory	hmac-sha1
Optional	hmac-sha224
Mandatory	hmac-sha256
Optional	hamc-sha384
Optional	hmac-sha512

SHA-1 truncated to 96 bits (12 octets) SHOULD be implemented.

3. Specifying Truncation

When space is at a premium and the strength of the full length of an HMAC is not needed, it is reasonable to truncate the HMAC output and use the truncated value for authentication. HMAC SHA-1 truncated to 96 bits is an option available in several IETF protocols, including IPsec and TLS.

The TSIG RR [RFC2845] includes a "MAC size" field, which gives the size of the MAC field in octets. However, [RFC2845] does not specify what to do if this MAC size differs from the length of the output of HMAC for a particular hash function. Truncation is indicated by a MAC size less than the HMAC size, as specified below.

3.1. Truncation Specification

The specification for TSIG handling is changed as follows:

1. If "MAC size" field is greater than HMAC output length:

   This case MUST NOT be generated and, if received, MUST cause the packet to be dropped and RCODE 1 (FORMERR) to be returned.

2. If "MAC size" field equals HMAC output length:

   Operation is as described in [RFC2845], and the entire output HMAC output is present.

3. "MAC size" field is less than HMAC output length but greater than that specified in case 4, below:

   This is sent when the signer has truncated the HMAC output to an allowable length, as described in RFC 2104, taking initial octets and discarding trailing octets. TSIG truncation can only be to an integral number of octets. On receipt of a packet with truncation thus indicated, the locally calculated MAC is similarly truncated and only the truncated values are compared for authentication. The request MAC used when calculating the TSIG MAC for a reply is the truncated request MAC.

4. "MAC size" field is less than the larger of 10 (octets) and half the length of the hash function in use:

   With the exception of certain TSIG error messages described in RFC 2845, Section 3.2, where it is permitted that the MAC size be zero, this case MUST NOT be generated and, if received, MUST cause the packet to be dropped and RCODE 1 (FORMERR) to be returned. The size limit for this case can also, for the hash functions mentioned in this document, be stated as less than half the hash function length for hash functions other than MD5 and less than 10 octets for MD5.

4. TSIG Truncation Policy and Error Provisions

Use of TSIG is by mutual agreement between a resolver and server. Implicit in such "agreement" are criterion as to acceptable keys and algorithms and, with the extensions in this document, truncations. Note that it is common for implementations to bind the TSIG secret key or keys that may be in place at a resolver and server to particular algorithms. Thus, such implementations only permit the use of an algorithm if there is an associated key in place. Receipt of an unknown, unimplemented, or disabled algorithm typically results in a BADKEY error.

Local policies MAY require the rejection of TSIGs, even though they use an algorithm for which implementation is mandatory.

When a local policy permits acceptance of a TSIG with a particular algorithm and a particular non-zero amount of truncation, it SHOULD also permit the use of that algorithm with lesser truncation (a longer MAC) up to the full HMAC output.

Regardless of a lower acceptable truncated MAC length specified by local policy, a reply SHOULD be sent with a MAC at least as long as that in the corresponding request, unless the request specified a MAC length longer than the HMAC output.

Implementations permitting multiple acceptable algorithms and/or truncations SHOULD permit this list to be ordered by presumed strength and SHOULD allow different truncations for the same algorithm to be treated as separate entities in this list. When so implemented, policies SHOULD accept a presumed stronger algorithm and truncation than the minimum strength required by the policy.

If a TSIG is received with truncation that is permitted under Section 3 above but the MAC is too short for the local policy in force, an RCODE of 22 (BADTRUNC) MUST be returned.

5. IANA Considerations

This document (1) registers the new TSIG algorithm identifiers listed in Section 2 with IANA and (2) allocates the BADTRUNC RCODE 22 in Section 4 [RFC2845].

6. Security Considerations

For all of the message authentication code algorithms listed herein, those producing longer values are believed to be stronger; however, while there have been some arguments that mild truncation can strengthen a MAC by reducing the information available to an attacker, excessive truncation clearly weakens authentication by reducing the number of bits an attacker has to try to break the authentication by brute force [RFC2104].

Significant progress has been made recently in cryptanalysis of hash function of the types used herein, all of which ultimately derive from the design of MD4. While the results so far should not effect

HMAC, the stronger SHA-1 and SHA-256 algorithms are being made mandatory due to caution.

See the Security Considerations section of [RFC2845]. See also the Security Considerations section of [RFC2104] from which the limits on truncation in this RFC were taken.

7. Normative References

[FIPS180-2] "Secure Hash Standard", (SHA-1/224/256/384/512) US Federal Information Processing Standard, with Change Notice 1, February 2004.

[RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm ", RFC 1321, April 1992.

[RFC2104] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

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

[RFC2845] Vixie, P., Gudmundsson, O., Eastlake 3rd, D., and B. Wellington, "Secret Key Transaction Authentication for DNS (TSIG)", RFC 2845, May 2000.

[RFC3174] Eastlake 3rd, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001.

[RFC3874] Housley, R., "A 224-bit One-way Hash Function: SHA-224", RFC 3874, September 2004.

[RFC4634] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA)", RFC 4634, July 2006.

[STD13] Mockapetris, P., "Domain names - concepts and facilities", STD 13, RFC 1034, November 1987.

Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

8. Informative References.

[RFC2930] Eastlake 3rd, D., "Secret Key Establishment for DNS (TKEY RR)", RFC 2930, September 2000.

[RFC2931] Eastlake 3rd, D., "DNS Request and Transaction Signatures (SIG(0)s)", RFC 2931, September 2000.

[RFC3645] Kwan, S., Garg, P., Gilroy, J., Esibov, L., Westhead, J., and R. Hall, "Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)", RFC 3645, October 2003.

Author's Address

Donald E. Eastlake 3rd
Motorola Laboratories
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-786-7554 (w)
EMail: Donald.Eastlake@motorola.com

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).