Network Working Group
Request for Comments: 2085
Category: Standards Track

M. Oehler
NSA
R. Glenn
NIST
February 1997

HMAC-MD5 IP Authentication with Replay Prevention

Status of This Memo

Abstract

This document describes a keyed-MD5 transform to be used in conjunction with the IP Authentication Header [RFC-1826]. The particular transform is based on [HMAC-MD5]. An option is also specified to guard against replay attacks.

Table of Contents

1. Introduction

The Authentication Header (AH) [RFC-1826] provides integrity and authentication for IP datagrams. The transform specified in this document uses a keyed-MD5 mechanism [HMAC-MD5]. The mechanism uses the (key-less) MD5 hash function [RFC-1321] which produces a message digest. When combined with an AH Key, authentication data is produced. This value is placed in the Authentication Data field of the AH [RFC-1826]. This value is also the basis for the data integrity service offered by the AH protocol.

To provide protection against replay attacks, a Replay Prevention field is included as a transform option. This field is used to help prevent attacks in which a message is stored and re-used later, replacing or repeating the original. The Security Parameters Index (SPI) [RFC-1825] is used to determine whether this option is included in the AH.

Familiarity with the following documents is assumed: "Security Architecture for the Internet Protocol" [RFC-1825], "IP Authentication Header" [RFC-1826], and "HMAC-MD5: Keyed-MD5 for Message Authentication" [HMAC-MD5].

All implementations that claim conformance or compliance with the IP Authentication Header specification [RFC-1826] MUST implement this HMAC-MD5 transform.

1.1 Terminology

In this document, the words that are used to define the significance of each particular requirement are usually capitalized. These words are:

- MUST

This word or the adjective "REQUIRED" means that the item is an absolute requirement of the specification.

- SHOULD

This word or the adjective "RECOMMENDED" means that there might exist valid reasons in particular circumstances to ignore this item, but the full implications should be understood and the case carefully weighed before taking a different course.

1.2 Keys

The "AH Key" is used as a shared secret between two communicating parties. The Key is not a "cryptographic key" as used in a traditional sense. Instead, the AH key (shared secret) is hashed with the transmitted data and thus, assures that an intervening party cannot duplicate the authentication data.

Even though an AH key is not a cryptographic key, the rudimentary

concerns of cryptographic keys still apply. Consider that the algorithm and most of the data used to produce the output is known. The strength of the transform lies in the singular mapping of the key (which needs to be strong) and the IP datagram (which is known) to the authentication data. Thus, implementations should, and as

frequently as possible, change the AH key. Keys need to be chosen at random, or generated using a cryptographically strong pseudo-random generator seeded with a random seed. [HMAC-MD5]

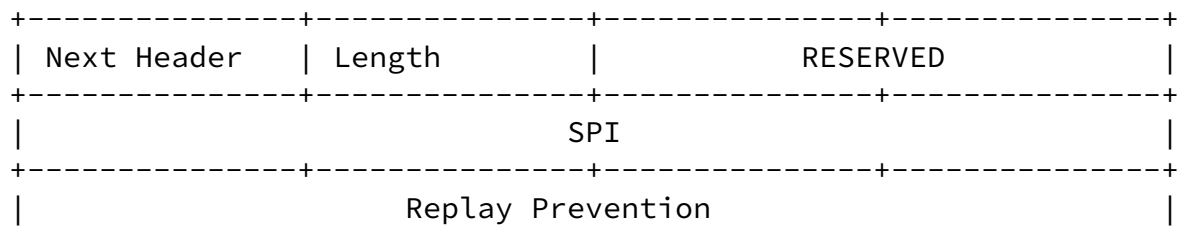
All conforming and compliant implementations MUST support a key length of 128 bits or less. Implementations SHOULD support longer key lengths as well. It is advised that the key length be chosen to be the length of the hash output, which is 128 bits for MD5. For other key lengths the following concerns MUST be considered.

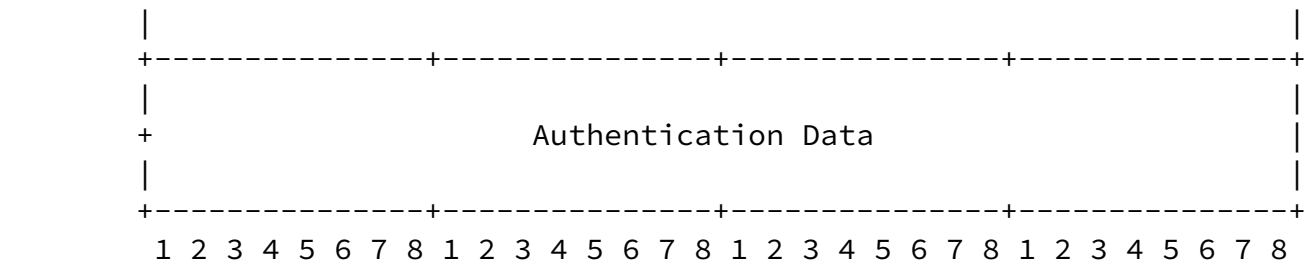
A key length of zero is prohibited and implementations MUST prevent key lengths of zero from being used with this transform, since no effective authentication could be provided by a zero-length key. Keys having a length less than 128 bits are strongly discouraged as it would decrease the security strength of the function. Keys longer than 128 bits are acceptable, but the extra length may not significantly increase the function strength. A longer key may be advisable if the randomness of the key is suspect. MD5 operates on 64-byte blocks. Keys longer than 64-bytes are first hashed using MD5. The resulting hash is then used to calculate the authentication data.

1.3 Data Size

MD5 produces a 128-bit value which is used as the authentication data. It is naturally 64 bit aligned and thus, does not need any padding for machines with native double words.

2. Packet Format





The Next Header, RESERVED, and SPI fields are specified in [RFC-1826]. The Length field is the length of the Replay Prevention field and the Authentication Data in 32-bit words.

2.1 Replay Prevention

The Replay Prevention field is a 64-bit value used to guarantee that each packet exchanged between two parties is different. Each IPsec Security Association specifies whether Replay Prevention is used for that Security Association. If Replay Prevention is NOT in use, then the Authentication Data field will directly follow the SPI field.

The 64-bit field is an up counter starting at a value of 1.

The secret shared key must not be used for a period of time that allows the counter to wrap, that is, to transmit more than 2^64 packets using a single key.

Upon receipt, the replay value is assured to be increasing. The implementation may accept out of order packets. The number of packets to accept out of order is an implementation detail. If an "out of order window" is supported, the implementation shall ensure that any and all packets accepted out of order are guaranteed not to have arrived before. That is, the implementation will accept any packet at most once.

When the destination address is a multicast address, replay protection is in use, and more than one sender is sharing the same IPsec Security Association to that multicast destination address, then Replay Protection SHOULD NOT be enabled. When replay protection is desired for a multicast session having multiple senders to the same multicast destination address, each sender SHOULD have its own IPsec Security Association.

[ESP-DES-MD5] provides example code that implements a 32 packet replay window and a test routine to show how it works.

2.2 Authentication Data Calculation

The authentication data is the output of the authentication algorithm (MD5). This value is calculated over the entire IP datagram. Fields within the datagram that are variant during transit and the authentication data field itself, must contain all zeros prior to the computation [RFC-1826]. The Replay Prevention field if present, is included in the calculation.

The definition and reference implementation of MD5 appears in [RFC-1321]. Let 'text' denote the data to which HMAC-MD5 is to be applied and K be the message authentication secret key shared by the parties. If K is longer than 64-bytes it MUST first be hashed using MD5. In this case, K is the resulting hash.

We define two fixed and different strings ipad and opad as follows (the 'i' and 'o' are mnemonics for inner and outer):

```
          ipad = the byte 0x36 repeated 64 times
          opad = the byte 0x5C repeated 64 times.
To compute HMAC-MD5 over the data `text' we perform
          MD5(K XOR opad, MD5(K XOR ipad, text))
Namely,
```

(1) append zeros to the end of K to create a 64 byte string (e.g., if K is of length 16 bytes it will be appended with 48 zero bytes 0x00)
(2) XOR (bitwise exclusive-OR) the 64 byte string computed in step (1) with ipad
(3) append the data stream 'text' to the 64 byte string resulting from step (2)
(4) apply MD5 to the stream generated in step (3)
(5) XOR (bitwise exclusive-OR) the 64 byte string computed in step (1) with opad
(6) append the MD5 result from step (4) to the 64 byte string resulting from step (5)
(7) apply MD5 to the stream generated in step (6) and output the result

This computation is described in more detail, along with example code and performance improvements, in [HMAC-MD5]. Implementers should consult [HMAC-MD5] for more information on this technique for keying a cryptographic hash function.

3. Security Considerations

The security provided by this transform is based on the strength of MD5, the correctness of the algorithm's implementation, the security of the key management mechanism and its implementation, the strength of the associated secret key, and upon the correctness of the implementations in all of the participating systems. [HMAC-MD5] contains a detailed discussion on the strengths and weaknesses of MD5.

Acknowledgments

This document is largely based on text written by Hugo Krawczyk. The format used was derived from work by William Simpson and Perry Metzger. The text on replay prevention is derived directly from work by Jim Hughes.

References

[RFC-1825] Atkinson, R., "Security Architecture for the Internet Protocol", RFC 1852, Naval Research Laboratory, July 1995.

[RFC-1826] Atkinson, R., "IP Authentication Header", RFC 1826, August 1995.

[RFC-1828] Metzger, P., and W. Simpson, "IP Authentication using Keyed MD5", RFC 1828, August 1995.

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

[HMAC-MD5] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[ESP-DES-MD5] Hughes, J., "Combined DES-CBC, MD5, and Replay

Prevention Security Transform", Work in Progress.

Authors' Addresses

Michael J. Oehler
National Security Agency
Atn: R23, INFOSEC Research and Development
9800 Savage Road
Fort Meade, MD 20755

EMail: mjo@tycho.ncsc.mil

Robert Glenn
NIST
Building 820, Room 455
Gaithersburg, MD 20899

EMail: rob.glenn@nist.gov