Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 4, 2013

D. Zhang
Huawei
M. Bhatia
Alcatel-Lucent
V. Manral
Hewlett-Packard Co.
July 03, 2012

Authenticating BFD using HMAC-SHA-2 procedures
draft-ietf-bfd-hmac-sha-01

Abstract

This document describes the mechanism to authenticate Bidirectional Forwarding Detection (BFD) protocol packets using Hashed Message Authentication Mode (HMAC) with the SHA-256, SHA-384, and SHA-512 algorithms. The mechanism described uses the Generic Cryptographic Authentication and Generic Meticulous Cryptographic Authentication sections to carry the authentication data. This document updates, but does not supercede, the cryptographic authentication mechanism specified in RFC 5880.

Requirements Language

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].

Status of this Memo

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

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on January 4, 2013.

Copyright 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
2. Cryptographic Aspects 3
3. Procedures at the Sending Side 5
4. Procedure at the Receiving Side 5
5. IANA Considerations 6
6. Security Considerations 6
7. References 7
7.1. Normative References 7
7.2. Informative References 8
Authors' Addresses 9

1. Introduction

The cryptographic authentication mechanisms specified in BFD [RFC5880] defines MD5 [RFC1321] and Secure Hash Algorithm (SHA-1) algorithms to authenticate BFD packets. The recent escalating series of attacks on MD5 and SHA-1 [SHA-1-attack1] [SHA-1-attack2] raise concerns about their remaining useful lifetime [RFC6151] [RFC6194].

These attacks may not necessarily result in direct vulnerabilities for Keyed-MD5 and Keyed-SHA-1 digests as message authentication codes because the colliding message may not correspond to a syntactically correct BFD protocol packet. Regardless, there is a need felt to deprecate MD5 and SHA-1 as the basis for the HMAC algorithm in favor of stronger digest algorithms.

This document adds support for Secure Hash Algorithms (SHA) defined in the US NIST Secure Hash Standard (SHS), which is defined by NIST FIPS 180-2 [FIPS-180-2]. [FIPS-180-2] includes SHA-1, SHA-224, SHA-256, SHA-384, and SHA-512. The HMAC authentication mode defined in NIST FIPS 198 is used [FIPS-198].

It is believed that [RFC2104] is mathematically identical to [FIPS-198] and it is also believed that algorithms in [RFC6234] are mathematically identical to [FIPS-180-2].

It should be noted that if SHA-1 is used in the HMAC construction then collision attacks currently known against SHA-1 do not apply. The new attacks on SHA-1 have no impact on the security of HMAC-SHA-1. NIST will be supporting HMAC-SHA-1 even after 2010 [NIST-HMAC-SHA] , whereas it would be dropping support for SHA-1 in digital signatures.

[I-D.ietf-bfd-generic-crypto-auth] defines new authentication types - Generic Cryptographic Authentication and Generic Meticulous Cryptographic Authenticationan extension that can be used for carrying the authentication digests defined in this document.

Implementations of this specification must include support for at least HMAC-SHA-256 and may include support for either of HMAC-SHA-384 or HMAC-SHA-512.

2. Cryptographic Aspects

In the algorithm description below, the following nomenclature, which is consistent with [FIPS-198], is used:

H is the specific hashing algorithm (e.g. SHA-256).

K is the password for the BFD packet.

Ko is the cryptographic key used with the hash algorithm.

B is the block size of H, measured in octets rather than bits. Note that B is the internal block size, not the hash size. For SHA-1 and SHA-256: B == 64 For SHA-384 and SHA-512: B == 128 L is the length of the hash, measured in octets rather than bits.

XOR is the exclusive-or operation.

Opad is the hexadecimal value 0x5c repeated B times.

Ipad is the hexadecimal value 0x36 repeated B times.

Apad is the hexadecimal value 0x878FE1F3 repeated (L/4) times.

(1) Preparation of the Key

In this application, Ko is always L octets long.

If the Authentication Key (K) is L octets long, then Ko is equal to K. If the Authentication Key (K) is more than L octets long, then Ko is set to H(K). If the Authentication Key (K) is less than L octets long, then Ko is set to the Authentication Key (K) with zeros appended to the end of the Authentication Key (K) such that Ko is L octets long.

(2) First Hash

First, the Authentication Data field in the Generic Authentication Section is filled with the value Apad and the Authentication Type field is set to 6 or 7 depending upon which Authentication Type is being used. The Sequence Number field MUST be set to bfd.XmitAuthSeq.

Then, a first hash, also known as the inner hash, is computed as follows:

First-Hash = H(Ko XOR Ipad || (BFD Packet))

(3) Second Hash T

Then a second hash, also known as the outer hash, is computed as follows:

Second-Hash = H(Ko XOR Opad || First-Hash)

(4) Result

The resultant Second-Hash becomes the Authentication Data that is sent in the Authentication Data field of the BFD Authentication Section. The length of the Authentication Data field is always identical to the message digest size of the specific hash function H that is being used.

This also means that the use of hash functions with larger output sizes will also increase the size of BFD Packet as transmitted on the wire.

3. Procedures at the Sending Side

Before a BFD device sends a BFD packet out, the device needs to select an appropriate BFD SA from its local key table if a keyed digest for the packet is required. If no appropriate SA is avaliable, the BFD packet MUST be discarded.

If an appropriate SA is avaliable, the device then derives the key and the associated authentication algorithm (HMAC-SHA-256, HMAC-SHA-384 or HMAC-SHA-512) from the SA.

The device then start performing the operations illustrated in Section 2. Before the authentication data is computed, the device MUST fill the Auth Type and the Auth length . The Sequence Number field MUST be set to bfd.XmitAuthSeq.

The value of Auth Length in the generic authentication section is various according to different authentication algorithms being used. Specifically, the value is 40 for HMAC-SHA-256, 56 for HMAC-SHA-384 and 72 for HMAC- SHA-512.

The Key ID is then filled.

After that, the authentication data is computed as illustrated in Section 3.

The result of the authentication algorithm is placed in the Authentication data, following the Key ID.

4. Procedure at the Receiving Side

Upon receiving a BFD packet with an generic authentication section appended, the receiving device needs to find an appropriate BFD SA from its local key table to verify the packet. The SA is located by

the Key ID in the authentication section of the packet.

If there is no SA is associated with the Key ID, the received packet MUST be discarded.

If bfd.AuthSeqKnown is 1, examine the Sequence Number field. For Cryptographic Authentication, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32 bit circular number space), the received packet MUST be discarded. For Meticulous Cryptographic Authentication, if the Sequence Number lies outside of the range of bfd.RcvAuthSeq+1 to bfd.RcvAuthSeq+(3*Detect Mult) inclusive (when treated as an unsigned 32 bit circular number space, the received packet MUST be discarded.

Authentication Algorithm dependent processing, needs to be performed, using the algorithm specified by the appropriate BFD SA for the received packet.

Before the device performs any processing, it needs to save the values of the Authentication Value field.

The device then needs to set the Authentication Value field with Apad before the authentication data is computed. The calculated data is compared with the received authentication data in the packet.

The packet MUST be discarded if the calculated data and the received authentication data do not match each other. In such a case, an error event SHOULD be logged.

A BFD implementation MAY be in a transition mode where it includes CRYPTO_AUTH or the MET_CRYPTO_AUTH information in packets but never verifies it. This is provided as a transition aid for networks in the process of migrating to the new CRYPTO_AUTH and MET_CRYPTO_AUTH based authentication schemes.

5. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

6. Security Considerations

The approach described in this document enhances the security of the

BFD protocol by adding, to the existing BFD cryptographic authentication methods, support for the SHA-2 algorithms defined in the NIST Secure Hash Standard (SHS) using the HMAC mode. However, the confidentiality protection for BFD packets is out of scope of this work .

Because all of the currently specified algorithms use symmetric cryptography, one cannot authenticate precisely which BFD device sent a given packet. However, one can authenticate that the sender knew the BFD Security Association (including the BFD SA's parameters) currently in use.

To enhance system security, the applied keys should be changed periodically and implementations SHOULD be able to store and use more than one key at the same time. The quality of the security provided by the cryptographic authentication option depends completely on the strength of the cryptographic algorithm and cryptographic mode in use, the strength of the key being used, and the correct implementation of the security mechanism in all communicating BFD implementations. Accordingly, the use of high assurance development methods is recommended. It also requires that all parties maintain the secrecy of the shared secret key. [RFC4086] provides guidance on methods for generating cryptographically random bits.

The value Apad is used here primarily for consistency with IETF specifications for HMAC-SHA authentication for RIPv2 [RFC4822], IS-IS [RFC5310] and OSPFv2 [RFC5709].

7. References

7.1. Normative References

[FIPS-180-2]
National Institute of Standards and Technology, FIPS PUB 180-2, "The Keyed-Hash Message Authentication Code (HMAC)", August 2002.

[FIPS-198]
National Institute of Standards and Technology, FIPS PUB 198, "The Keyed-Hash Message Authentication Code (HMAC)", March 2002.

[I-D.ietf-bfd-generic-crypto-auth]
Bhatia, M., Manral, V., and D. Zhang, "BFD Generic Cryptographic Authentication", draft-ietf-bfd-generic-crypto-auth-02 (work in progress), June 2012.

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

[RFC6039] Manral, V., Bhatia, M., Jaeggli, J., and R. White, "Issues with Existing Cryptographic Protection Methods for Routing Protocols", RFC 6039, October 2010.

[RFC6151] Turner, S. and L. Chen, "Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms", RFC 6151, March 2011.

[RFC6194] Polk, T., Chen, L., Turner, S., and P. Hoffman, "Security Considerations for the SHA-0 and SHA-1 Message-Digest Algorithms", RFC 6194, March 2011.

7.2. Informative References

[Dobb96a] Dobbertin, H., "Cryptanalysis of MD5 Compress", May 1996.

[Dobb96b] Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes", 1996.

[I-D.ietf-karp-design-guide]
Lebovitz, G. and M. Bhatia, "Keying and Authentication for Routing Protocols (KARP) Design Guidelines", draft-ietf-karp-design-guide-10 (work in progress), December 2011.

[MD5-attack]
Wang, X., Feng, D., Lai, X., and H. Yu, "Collisions for Hash Functions MD4, MD5, HAVAL-128 and RIPEMD", August 2004.

[NIST-HMAC-SHA]
National Institute of Standards and Technology, Available online at http://csrc.nist.gov/groups/ST/hash/policy.html, "NIST's Policy on Hash Functions", 2006.

[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.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC4822] Atkinson, R. and M. Fanto, "RIPv2 Cryptographic Authentication", RFC 4822, February 2007.

[RFC5310] Bhatia, M., Manral, V., Li, T., Atkinson, R., White, R., and M. Fanto, "IS-IS Generic Cryptographic Authentication", RFC 5310, February 2009.

[RFC5709] Bhatia, M., Manral, V., Fanto, M., White, R., Barnes, M., Li, T., and R. Atkinson, "OSPFv2 HMAC-SHA Cryptographic Authentication", RFC 5709, October 2009.

[RFC5880] Katz, D. and D. Ward, "Bidirectional Forwarding Detection (BFD)", RFC 5880, June 2010.

[RFC6234] Eastlake, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, May 2011.

[SHA-1-attack1] Wang, X., Yin, Y., and H. Yu, "Finding Collisions in the Full SHA-1", 2005.

[SHA-1-attack2] Wang, X., Yao, A., and F. Yao, "New Collision Search for SHA-1", 2005.

Authors' Addresses

Dacheng Zhang
Huawei
Beijing,
China

Email: zhangdacheng@huawei.com

Manav Bhatia
Alcatel-Lucent
Bangalore 560045
India

Email: manav.bhatia@alcatel-lucent.com

Vishwas Manral
Hewlett-Packard Co.
19111 Pruneridge Ave.
Cupertino, CA 95014
USA

Email: vishwas.manral@hp.com