HMAC-SHA-256-128 Authentication Protocol in USM for SNMP
draft-hmac-sha-256-128-usm-snmp-00

Abstract

This memo specifies a new optional HMAC-SHA-256-128 authentication protocol for the User-based Security Model (USM) for SNMPv3 defined in RFC 3414.

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 May 09, 2014.

Copyright Notice

Copyright (c) 2013 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.
1. Introduction

The User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3) is specified in RFC 3414 [RFC3414]. Within USM, two different authentication protocols, HMAC-MD5-96 and HMAC-SHA-96, are defined based on the hash functions MD5 and SHA-1, respectively. This memo specifies a new HMAC-SHA-256-128 authentication protocol for USM using an HMAC based on the SHA-256 hash function [SHA] and truncated to 128 bits. The protocol is a straightforward adaptation of the authentication protocols HMAC-MD5-96 and HMAC-SHA-96 to the SHA-256 based HMAC. The use and support of the HMAC-SHA-256-128 authentication protocol is OPTIONAL.

2. The HMAC-SHA-256-128 Authentication Protocol

This section describes the HMAC-SHA-256-128 authentication protocol. This protocol uses the SHA-256 hash function which is described in [SHA], in HMAC mode described in [RFC2104], truncating the output to 128 bits. Source code for SHA-256 and HMAC-SHA-256 (without truncation) can be found in [RFC4634]. Test vectors for HMAC-SHA-256 (without truncation) and HMAC-SHA-256-128 are given in [RFC4231] and in [RFC4868], respectively.

This protocol is identified by usmHMACSHA256AuthProtocol.

2.1. Deviations from the HMAC-SHA-96 Authentication Protocol

The HMAC-SHA-256-128 authentication protocol is a straightforward adaptation of the HMAC-MD5-96 and HMAC-SHA-96 authentication protocols. Precisely, it differs from the HMAC-MD5-96 and HMAC-SHA-96 authentication protocols in the following aspects:

- The SHA-256 hash function is used to compute the message digest in the HMAC computation according to [RFC2104], as opposed to the MD5
hash function [RFC1321] and SHA-1 hash function [SHA] used in HMAC-MD5-96 and HMAC-SHA-96, respectively. Consequently, the length of the message digest prior to truncation is 256 bits.

- The 256 bit message digest is truncated to 16 octets as opposed to the truncation to 12 octets in HMAC-MD5-96 and HMAC-SHA-96.
- The user’s secret key to be used when calculating a digest MUST be 32 octets long as opposed to the keys being 16 and 20 octets long in HMAC-MD5-96 and HMAC-SHA-96, respectively.

2.2. Processing

This section describes the procedures for the HMAC-SHA-256-128 authentication protocol. The descriptions are based on the definition of services and data elements defined for HMAC-SHA-96 in RFC 3414 [RFC3414] with the deviations listed in Section 2.1.

2.2.1. Processing an Outgoing Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an outgoing message using the usmHMACSHA256AuthProtocol.

1. The msgAuthenticationParameters field is set to serialization, according to the rules in [RFC3417], of an OCTET STRING containing 16 zero octets.

2. From the secret authKey, two keys K1 and K2 are derived:
   a) extend the authKey to 64 octets by appending 32 zero octets; save it as extendedAuthKey;
   b) obtain IPAD by replicating the octet 0x36 64 times;
   c) obtain K1 by XORing extendedAuthKey with IPAD;
   d) obtain OPAD by replicating the octet 0x5C 64 times;
   e) obtain K2 by XORing extendedAuthKey with OPAD.

3. Prepend K1 to the wholeMsg and calculate the SHA-256 digest over it according to [SHA].

4. Prepend K2 to the result of the previous step and calculate the SHA-256 digest over it according to [SHA]. Take the first 16 octets of the final digest – this is the Message Authentication Code (MAC).
5. Replace the msgAuthenticationParameters field with the MAC obtained in the previous step.

6. The authenticatedWholeMsg is then returned to the caller together with statusInformation indicating success.

2.2.2. Processing an Incoming Message

This section describes the procedure followed by an SNMP engine whenever it must authenticate an incoming message using the usmHMACSHA256AuthProtocol.

1. If the digest received in the msgAuthenticationParameters field is not 16 octets long, then a failure and an errorIndication (authenticationError) is returned to the calling module.

2. The MAC received in the msgAuthenticationParameters field is saved.

3. The digest in the msgAuthenticationParameters field is replaced by the 16 zero octets.

4. From the secret authKey, two keys K1 and K2 are derived:
   a) extend the authKey to 64 octets by appending 32 zero octets; save it as extendedAuthKey
   b) obtain IPAD by replicating the octet 0x36 64 times;
   c) obtain K1 by XORing extendedAuthKey with IPAD;
   d) obtain OPAD by replicating the octet 0x5C 64 times;
   e) obtain K2 by XORing extendedAuthKey with OPAD.

5. The MAC is calculated over the wholeMsg:
   a) prepend K1 to the wholeMsg and calculate the SHA-256 digest over it;
   b) prepend K2 to the result of step 5.a and calculate the SHA-256 digest over it;
   c) first 16 octets of the result of step 5.b is the MAC.

The msgAuthenticationParameters field is replaced with the MAC value that was saved in step 2.
6. The newly calculated MAC is compared with the MAC saved in step 2. If they do not match, then a failure and an errorIndication (authenticationFailure) are returned to the calling module.

7. The authenticatedWholeMsg and statusInformation indicating success are then returned to the caller.

3. Security Considerations

The security considerations of [RFC3414] also apply to the use of the HMAC-SHA-256-128 authentication protocol in SNMP. A general discussion of the security of the HMAC construction is given in [RFC2104].

4. IANA Considerations

IANA is requested to assign an OID for the usmHMACSHA256AuthProtocol module under the SnmpAuthProtocols subtree, maintained in the registry at http://www.iana.org/assignments/smi-numbers.

5. References

5.1. Normative References


5.2. Informative References


Authors’ Addresses

Johannes Merkle
secunet Security Networks
Mergenthaler Allee 77
65760 Eschborn
Germany
Phone: +49 201 5454 3091
EMail: johannes.merkle@secunet.com

Manfred Lochter
BSI
Postfach 200363
53133 Bonn
Germany
Phone: +49 228 9582 5643
EMail: manfred.lochter@bsi.bund.de