Internet Draft
<draft-ietf-ospf-hmac-sha-07.txt>
Category: Standards-Track
Expires: 31 Jan 2010
Updates: RFC 2328

M. Bhatia
Alcatel-Lucent
V. Manral
IP Infusion
M. Fanto
Aegis Data Security
R. White
Cisco Systems
T. Li
Ericsson
M. Barnes
Cisco Systems
R. Atkinson
Extreme Networks

31 August 2009

OSPFv2 HMAC-SHA Cryptographic Authentication
<draft-ietf-ospf-hmac-sha-07.txt>

Status of this Memo

Abstract

This document describes how the NIST Secure Hash Standard family of algorithms can be used with OSPF version 2's built-in cryptographic authentication mechanism. This updates, but does not supercede, the cryptographic authentication mechanism specified in RFC 2328.

1. INTRODUCTION

A variety of risks exist when depoying any routing protocol.[Bell89] This document provides an update to OSPFv2 Cryptographic Authentication, which is specified in Appendix D of RFC 2328. This document does not deprecate or supercede RFC 2328. OSPFv2 itself is defined in RFC 2328. [RFC 2328]

This document adds support for Secure Hash Algorithms defined in the US NIST Secure Hash Standard (SHS) as defined by NIST 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 [RFC 2104] is mathematically identical to [FIPS-198] and also believed that algorithms in [RFC 4684] are mathematically identical to [FIPS-180-2].

The creation of this addition to OSPFv2 was driven by operator requests that they be able to use the NIST SHS family of algorithms in the NIST HMAC mode, instead of being forced to use the Keyed-MD5 algorithm and mode with OSPFv2 Cryptographic Authentication. Cryptographic matters are discussed in more

detail in the Security Considerations section of this document.

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. [RFC 2119]

2. Background

All OSPF protocol exchanges can be authenticated. The OSPF packet header (see Section A.3.1 of RFC-2328) includes an Authentication Type field, and 64-bits of data for use by the appropriate authentication scheme (determined by the type field).

The authentication type is configurable on a per-interface (or equivalently, on a per-network/subnet) basis. Additional authentication data is also configurable on a per-interface basis.

OSPF Authentication types 0, 1, and 2 are defined by RFC 2328. This document provides an update to RFC 2328 that is only applicable to Authentication Type 2, "Cryptographic Authentication".

3. Cryptographic authentication with NIST SHS in HMAC mode

Using this authentication type, a shared secret key is configured in all routers attached to a common network/subnet. For each OSPF protocol packet, the key is used to generate/verify a "message digest" that is appended to the end of the OSPF packet. The message digest is a one-way function of the OSPF protocol packet and the secret key. Since the secret key is never sent over the network in the clear, protection is provided against passive attacks. [RFC 1704]

The algorithms used to generate and verify the message digest are specified implicitly by the secret key. This specification discusses the computation of OSPF Cryptographic Authentication data when any of the NIST SHS family of algorithms is used in the Hashed Message Authentication Code (HMAC) mode.
Please also see RFC 2328, Appendix D.

With the additions in this document, the currently valid algorithms (including mode) for OSPFv2 Cryptographic Authentication include:

Keyed-MD5 (defined in RFC-2328, Appendix D)

```
     HMAC-SHA-1          (defined here)
     HMAC-SHA-256        (defined here)
     HMAC-SHA-384        (defined here)
     HMAC-SHA-512        (defined here)
```

Of the above, implementations of this specification MUST include support for at least:

```
     HMAC-SHA-256
```

and SHOULD include support for:

```
     HMAC-SHA-1
```

and SHOULD also (for backwards compatibility with existing implementations and deployments) include support for:

```
     Keyed-MD5
```

and MAY also include support for:

```
        HMAC-SHA-384
        HMAC-SHA-512
```

An implementation of this specification MUST allow network operators to configure ANY authentication algorithm supported by that implementation for use with ANY given Key-ID value that is configured into that OSPFv2 router.

3.1. Generating Cryptographic Authentication

The overall cryptographic authentication process defined in Appendix D of RFC 2328 remains unchanged. However, the specific cryptographic details (i.e. SHA rather than MD5, HMAC rather than Keyed-Hash) are defined herein. To reduce the potential for confusion, this section minimises the repetition of text from RFC 2328, Appendix D, which is incorporated here by reference.[RFC 2328]

First, following the procedure defined in RFC 2328, Appendix D, select the appropriate OSPFv2 Security Association for use with this packet and set the Key-ID field to the KeyID value of that OSPFv2 Security Association.

Second, set the Authentication Type to cryptographic authentication, and set the Authentication Data Length field to the length (measured in bytes, not bits) of the cryptographic hash that will be used. When any NIST SHS algorithm is used in HMAC mode with OSPFv2 Cryptographic Authentication, the Authentication Data Length is equal to the normal hash output length (measured in bytes) for the specific NIST SHS algorithm in use. For example, with NIST SHA-256, the Authentication Data

Length is 32 bytes.

Third, The 32-bit Cryptographic sequence number is set in accordance with the procedures in RFC 2328, Appendix D applicable to the Cryptographic Authentication type.

Fourth, The message digest is then calculated and appended to the OSPF packet, as described below in Section 3.3. The KeyID, Authentication Algorithm, and Key to be used for calculating the digest are all components of the selected OSPFv2 Security Association. Input to the authentication algorithm consists of the OSPF packet and the secret key.

3.2 OSPFv2 Security Association

This document uses the term OSPFv2 Security Association (OSPFv2 SA) to refer to the authentication key information defined in Section D.3, pages 228 and 229, of RFC 2328. The OSPFv2 protocol does not include an in-band mechanism to create or manage OSPFv2 Security Associations. The parameters of an OSPFv2 Security Association are updated to be:

Key Identifier (KeyID)
: This is an 8-bit unsigned value used to uniquely identify an OSPFv2 SA and is configured either by the router administrator (or, in the future, possibly by some key management protocol specified by the IETF). The receiver uses this to locate the appropriate OSPFv2 SA to use. The sender puts this KeyID value in the OSPF packet based on the active OSPF configuration.

Authentication Algorithm
: This indicates the authentication algorithm (and also the cryptographic mode, such as HMAC) to be used. This information SHOULD never be sent over the wire in cleartext form. At present valid values are: Keyed-MD5, HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

Authentication Key
: This is the cryptographic key used for cryptographic authentication with this OSPFv2 SA. This value SHOULD never be

sent over the wire in cleartext form.
This is noted as "K" in Section 3.3 below.

Key Start Accept
The time that this OSPF router will accept packets that have been created with this OSPF Security Association.

Key Start Generate
The time that this OSPF router will begin using this OSPF Security Association for OSPF packet generation.

Key Stop Generate
The time that this OSPF router will stop using this OSPF Security Association for OSPF packet generation.

Key Stop Accept
The time that this OSPF router will stop accepting packets generated with this OSPF Security Association.

In order to achieve smooth key transition, KeyStartAccept SHOULD be less than KeyStartGenerate and KeyStopGenerate SHOULD be less than KeyStopAccept. If KeyStopGenerate and KeyStopAccept are left unspecified, the key's lifetime is infinite. When a new key replaces an old, the KeyStartGenerate time for the new key MUST be less than or equal to the KeyStopGenerate time of the old key.

Key storage SHOULD persist across a system restart, warm or cold, to avoid operational issues. In the event that the last key associated with an interface expires, it is unacceptable to revert to an unauthenticated condition, and not advisable to disrupt routing. Therefore, the router should send a "last authentication key expiration" notification to the network manager and treat the key as having an infinite lifetime until the lifetime is extended, the key is deleted by network management, or a new key is configured.

3.3 Cryptographic Aspects

This describes the computation of the Authentication Data value when any NIST SHS algorithm is used in the HMAC mode with OSPFv2 Cryptographic Authentication.

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 authentication key for the OSPFv2 security
           association.
      Ko     is the cryptographic key used with the hash algorithm.
      B    is the block size of H, measured in octets,
           rather than bits.  Note well 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.

      Implementation note:
           This definition of Apad means that Apad always
        is the same length as the hash output.

   (1) PREPARATION OF 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 OSPFv2 packet's Authentication Trailer, which
     is the appendage described in RFC 2328, Section D.4.3,
     Page 233, items (6)(a) and (6)(d), is filled with the value
     Apad, and the Authentication Type field is set to 2.

     Then, a first hash, also known as the inner hash, is computed
     as follows:
           First-Hash = H(Ko XOR Ipad || (OSPFv2 Packet))

     Implementation Notes:
       Note that the First-Hash above includes the Authentication
       Trailer containing the Apad value, as well as the OSPF
       packet, as per RFC 2328, Section D.4.3.

       The definition of Apad (above) ensures it is always the same
       length as the hash output.  This is consistent with RFC 2328.
       The "(OSPFv2 Packet)" mentioned in the First Hash (above)
```

does include the OSPF Authentication Trailer.

The digest length for SHA-1 is 20 bytes, for SHA-256 is 32 bytes, for SHA-384 is 48 bytes, and for SHA-512 is 64-bytes.

(3) SECOND HASH

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 result Second-Hash becomes the Authentication Data that is sent in the Authentication Trailer of the OSPFv2 packet. The length of the Authentication Trailer 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 the OSPFv2 packet as transmitted on the wire.

Implementation Note:

RFC 2328, Appendix D specifies that the Authentication Trailer is not counted in the OSPF packet's own length field, but is included in the packet's IP length field.

3.4. Message verification

Message verification follows the procedure defined in RFC 2328, except that the cryptographic calculation of the message digest follows the procedure in Section 3.3 above when any NIST SHS algorithm in the HMAC mode is in use. Kindly recall that the cryptographic algorithm/mode in use is indicated implicitly by the Key-ID of the received OSPFv2 packet.

Implementation Notes:

One must save the received digest value before calculating the expected digest value, so that after that calculation the received value can be compared with the expected value to determine whether to accept that OSPF packet.

RFC 2328, Section D.4.3 (6) (c) should be read very closely prior to implementing the above. With SHA algorithms in HMAC mode, Apad is placed where the MD5 key would be put if Keyed-MD5 were in use.

3.5 Changing OSPFv2 Security Associations

Using KeyIDs makes changing the active OSPFv2 SA convenient. An implementation can choose to associate a lifetime with each OSPFv2 SA and can thus automatically switch to a different OSPFv2 SA based on the lifetimes of the configured OSPFv2 SA(s).

After changing the active OSPFv2 SA, the OSPF sender will use the (different) KeyID value associated with the newly active OSPFv2 SA. The receiver will use this new KeyID to select the appropriate (new) OSPFv2 SA to use with the received OSPF packet containing the new KeyID value.

Because the KeyID field is present, the receiver does not need to try all configured OSPFv2 Security Associations with any received OSPFv2 packet. This can mitigate some of the risks of a Denial-of-Service attack on the OSPF instance, but does not entirely prevent all conceivable DoS attacks. For example, an on-link adversary still could generate OSPFv2 packets that are synactically valid, but contain invalid Authentication Data, thereby forcing the receiver(s) to perform expensive cryptographic computations to discover that the packets are invalid.

4. Security Considerations

This document enhances the security of the OSPFv2 routing protocol by adding support for the algorithms defined in the NIST Secure Hash Standard (SHS) using the Hashed Message Authentication Code (HMAC) mode to the existing OSPFv2 Cryptographic Authentication method, and support for the Hashed Message Authentication Code (HMAC) mode.

This provides several alternatives to the existing Keyed-MD5 mechanism. There are published concerns about the overall strength of the MD5 algorithm. [Dobb96a, Dobb96b, Wang04] While those published concerns apply to the use of MD5 in other modes (e.g. use of MD5 X.509v3/PKIX digital certificates), they are not an attack upon Keyed-MD5, which is what OSPFv2 specified in RFC 2328. There are also published concerns about the SHA algorithm [Wang05] and also concerns about the MD5 and SHA algorithms in the HMAC mode [RR07, RR08]. Separately, some organisations (e.g. US Government) prefer NIST algorithms, such as the SHA family, over other algorithms for local policy reasons.

The value Apad is used here primarily for consistency with

IETF specifications for HMAC-SHA authentication of RIPv2 SHA [RFC 4822] and IS-IS SHA [RFC 5310] and to minimise OSPF protocol processing changes in Section D.4.3 of RFC 2328. [RFC 2328]

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 OSPF 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. [RFC 4086] provides guidance on methods for generating cryptographically random bits.

This mechanism is vulnerable to a replay attack by any on-link node. An on-link node could record a legitimate OSPF packet sent on the link, then replay that packet at the next time the recorded OSPF packet's sequence number is valid. This replay attack could cause significant routing disruptions within the OSPF domain.

Ideally, for example to prevent the preceding attack, each OSPF Security Association would be replaced by a new and different OSPF Security Association before any sequence number were reused. As of the date this document was published, no form of automated key management has been standardised for OSPF. So, as of the date this document was published, common operational practice has been to use the same OSPF authentication key for very long periods of time. This operational practice is undesirable for many reasons. Therefore, it is clearly desirable to develop and standardise some automated key management mechanism for OSPF.

Because all of the currently specified algorithms use symmetric cryptography, one cannot authenticate precisely which OSPF router sent a given packet. However, one can authenticate that the sender knew the OSPF Security Association (including the OSPFv2 SA's parameters) currently in use.

Because a routing protocol contains information that need not be kept secret, privacy is not a requirement. However, authentication of the messages within the protocol is of interest, to reduce the risk of an adversary compromising the routing system by deliberately injecting false

information into the routing system.

The technology in this document enhances an authentication mechanism for OSPFv2. The mechanism described here is not perfect and need not be perfect. Instead, this mechanism represents a significant increase in the work function of an adversary attacking OSPFv2, as compared with plain-text authentication or null authentication, while not causing undue implementation, deployment, or operational complexity. Denial of service attacks are not generally preventable in a useful networking protocol. [VK83]

Because of implementation considerations, including the need for backwards compatibility, this specification uses the same mechanism as specified in RFC 2328 and limits itself to adding support for additional cryptographic hash functions. Also, some large network operators have indicated they prefer to retain the basic mechanism defined in RFC 2328, rather than migrate to IP Security, due to deployment and operational considerations. If all the OSPFv2 routers supported IPsec, then IPsec tunnels could be used in lieu of this mechanism.[RFC 4301] This would, however, relegate the topology to point-to-point adjacencies over the mesh of IPsec tunnels.

If a stronger authentication were believed to be required, then the use of a full digital signature [RFC 2154] would be an approach that should be seriously considered. Use of full digital signatures would enable precise authentication of the OSPF router originating each OSPF link-state advertisement, and thereby provide much stronger integrity protection for the OSPF routing domain.

5. IANA CONSIDERATIONS

The OSPF Authentication Codes registry entry for Cryptographic Authentication (Registry Code 2) must be updated to refer to this document as well as RFC 2328.

6. ACKNOWLEDGEMENTS

The authors would like to thank Bill Burr, Tim Polk, John Kelsey, and Morris Dworkin of (US) NIST for review of portions of this document that are directly derived from the closely related work on RIPv2 Cryptographic Authentication [RFC 4822].

David Black, Nevil Brownlee, Acee Lindem, and Hilarie Orman (in alphabetical order by last name) provided feedback on earlier

versions of this document. That feedback has greatly improved both the technical content and the readability of the current draft.

Henrik Levkowetz's Internet Draft tools were very helpful in preparing this draft and are much appreciated.

7. REFERENCES

7.1 Normative References

[FIPS-180-2] US National Institute of Standards & Technology, "Secure Hash Standard (SHS)", FIPS PUB 180-2, August 2002.

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

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

[RFC 2328] Moy, J., "OSPF Version 2", RFC 2328, April 1998.

7.2 Informative References

[Bell89] S. Bellovin, "Security Problems in the TCP/IP Protocol Suite", ACM Computer Communications Review, Volume 19, Number 2, pp. 32-48, April 1989.

[Dobb96a] Dobbertin, H, "Cryptanalysis of MD5 Compress", Technical Report, 2 May 1996. (Presented at the Rump Session of EuroCrypt 1996.)

[Dobb96b] Dobbertin, H, "The Status of MD5 After a Recent Attack", CryptoBytes, Vol. 2, No. 2, Summer 1996.

[RFC 1704] N. Haller and R. Atkinson, "On Internet Authentication", RFC 1704, October 1994.

[RFC 2104] Krawczyk, H. et alia, "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997.

[RFC 2154] Murphy, S., Badger, M. and B. Wellington, "OSPF with Digital Signatures", RFC 2154, June 1997.

[RFC 4086] Eastlake, D., 3rd, Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP-106, RFC 4086, June 2005.

[RFC 4301] Kent, S. & K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC 4684] Eastlake 3rd, D., & T. Hansen, "US Secure Hash Algorithms (SHA and HMAC-SHA)", RFC 4634, July 2006.

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

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

[RR07] Rechberger, Christian & Vincent Rijmen, "On Authentication with HMAC and Non-random Properties", Financial Cryptography and Data Security, Lecture Notes in Computer Science, Volume 4886/2008, Springer-Verlag, Berlin, December 2007.

[RR08] Rechberger, Christian & Vincent Rijmen, "New Results on NMAC/HMAC when Instantiated with Popular Hash Functions", Journal of Universal Computer Science, Volume 14, Number 3, pp. 347-376, 1 February 2008.

[VK83] Voydock, V. and S. Kent, "Security Mechanisms in High-level Networks", ACM Computing Surveys, Vol. 15, No. 2, June 1983.

[Wang04] Wang, X. et alia, "Collisions for Hash Functions MD4, MD5, HAVAL-128, and RIPEMD", August 2004, IACR. http://eprint.iacr.org/2004/199

[Wang05] Wang, X. et alia, "Finding Collisions in the Full SHA-1" Proceedings of Crypto 2005, Lecture Notes in Computer Science, Volume 3621, pp. 17-36, Springer-Verlag, Berlin, August 31, 2005.

AUTHORS

Manav Bhatia
Alcatel-Lucent
Bangalore,
India

EMail: manav@alcatel-lucent.com

Vishwas Manral
IP Infusion
Almora, Uttarakhand
India

EMail: vishwas@ipinfusion.com

Matthew J. Fanto
Aegis Data Security
Dearborn, MI
USA

EMail: mfanto@aegisdatasecurity.com

Russ I. White
Cisco Systems
7025 Kit Creek Road
P.O. Box 14987
RTP, NC
27709 USA

EMail: riw@cisco.com

Tony Li
Ericsson
300 Holger Way
San Jose, CA
95134 USA

Email: tony.li@tony.li

M. Barnes
Cisco Systems
225 West Tasman Drive
San Jose, CA
95134 USA

Email: mjbarnes@cisco.com

Randall J. Atkinson

Extreme Networks
3585 Monroe Street
Santa Clara, CA
95051 USA

Phone: +1 (408) 579-2800
EMail: rja@extremenetworks.com

Expires: 31 JAN 2010