TCPM Working Group R. Bonica
Internet-Draft Juniper Networks
Expires: August 17, 2007 B. Weis
S. Viswanathan
Cisco Systems
A. Lange
Alcatel
O. Wheeler
BT

February 13, 2007

Authentication for TCP-based Routing and Management Protocols
draft-bonica-tcp-auth-06

Status of this Memo

Copyright Notice

Abstract

This memo describes a TCP extension that enhances security for BGP, LDP and other TCP-based protocols. It is intended for applications

where secure administrative access to both the end-points of the TCP connection is normally available. TCP peers can use this extension to authenticate messages passed between one another.

The strategy described herein improves upon current practice, which is described in RFC 2385. Using this new strategy, TCP peers can update authentication keys during the lifetime of a TCP connection. TCP peers can also use stronger authentication algorithms to authenticate routing messages.

Table of Contents

1. Conventions Used In 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 RFC2119 [1].

2. Terminology

The following terms are used in this document:

key - A data structure used to authenticate TCP segments. One or more keys can be associated with a TCP connection. Each key contains an identifier, a shared secret, an algorithm identifier, an "active flag" and an "eligible flag".

key set - A set of keys that is associated with a TCP connection. A single key set can be associated with multiple TCP connections. Each key within a key set contains an identifier that is unique within the key set.

active key - Each key set contains exactly one active key. The sending TCP station uses the shared secret from its active key to generate a Message Authentication Code (MAC) for outgoing TCP segments. The "active flag" on a key indicates whether a particular key is active.

eligible key - Each key set contains zero or more eligible keys. The receiving TCP station uses the shared secret from a key to authenticate an incoming TCP segment only if that key is eligible. The "eligible flag" on a key indicates whether a particular key is eligible.

3. Introduction

RFC 2385 [8] proposes a mechanism that authenticates TCP [2] sessions by including a message authentication code (MAC) in each TCP header. Authentication coverage includes the following fields:

- the TCP pseudo-header

- the TCP header, excluding options, and assuming a checksum of zero

- the TCP segment data (if any)

To spoof a connection using the scheme described above, an attacker would not only have to guess TCP sequence numbers, but would also have to obtain the key that was used to calculate the MAC. This key never appears in the connection stream.

RFC 3562 [9] addresses key management considerations regarding the TCP MD5 Signature Option. Based upon the strength of the MD5 [10] hashing algorithm, RFC 3562 recommends that keys be changed at least every 90 days.

Unfortunately, the strategy described in RFC 2385 permits keys to be changed during the lifetime of a TCP connection only so long as the change is synchronized at both ends. This limitation has proven to be a significant deterrent to the effective deployment of the TCP MD5 Signature Option. This memo addresses that limitation.

Also, the MD5 algorithm does not now provide a sufficient level of security, and recently published attacks motivate its replacement. In addition, the keyed hash MAC construction used by RFC 2385 has serious cryptographic weaknesses. An attacker who can find a collision in the underlying hash function can forge a MAC using a simple chosen-message attack [11]. This memo makes use of MAC algorithms that do not have these weaknesses. It also provides a mechanism to add additional algorithms as the state-of-the-art in cryptography progresses.

4. Proposal

This memo proposes a TCP Enhanced Authentication Option that is used as follows:

Network operators associate a set of keys with each protected TCP connection. Each key contains an identifier that is unique to the key set, a shared secret, an algorithm identifier, an "active flag" and an "eligible flag".

Whenever TCP generates a segment, it searches the associated key set for its active key. Each key set MUST have exactly one active key that is identified as such by having its "active flag" set.

Having identified the active key, TCP executes the following sequence:

- append the TCP Enhanced Authentication Option to the TCP header.
(See Section 6 of this document for details regarding the TCP Enhanced Authentication Option.)

- update the TCP Enhanced Authentication Option to include the active key's unique identifier

- calculate a Message Authentication Code (MAC) using the shared secret from the active key. (See Section 8 of this document for MAC calculation details.)

- update the TCP Enhanced Authentication Option to include the MAC that was calculated above

- calculate and update the TCP checksum

- forward the segment to a TCP peer.

The receiving TCP associates the inbound TCP segment with a local key set based upon source IP address, destination IP address, source port and destination port. It then searches the associated key set for a key whose identifier matches that which was specified by the incoming segment option.

If TCP finds such a key and if that key's "eligible flag" is set, TCP continues processing. If no matching eligible key is found then TCP MUST declare an authentication failure and discard the segment.

TCP verifies that the algorithm used to produce the MAC is correct. The verification is done by comparing the algorithm attribute associated with the key with the algorithm id listed in the segment option. If the algorithm identifiers do not match, then the MAC calculation will fail. TCP MUST declare an authentication error and discard the segment.

TCP uses the shared secret from the key to calculate a MAC. TCP will accept the segment if the calculated MAC matches the MAC specified by the inbound segment. Otherwise, TCP MUST declare an authentication failure and discard the segment.

TCP MUST also declare an authentication failure and discard a segment if the segment is received from a connection that is associated with a key set and the segment does not include the TCP Enhanced

Authentication Option.

To help protect against denial of service attacks it is RECOMMENDED that the inbound TCP segment is validated against the normal TCP criteria (e.g. that the segment sequence number is within the current

receive window) prior to the MAC being calculated. Inbound TCP segments on connections requiring the Enhanced Authentication Option MUST NOT cause the current TCP state variables for that connection to be updated unless the MAC has been verified as correct.

An authentication failure MUST NOT produce any response back to the sender. Routers SHOULD log authentication failures.

Unlike other TCP extensions (e.g., the Window Scale option [12]), the absence of the option in the SYN,ACK segment must not cause the sender to disable its sending of authentication data. This negotiation is typically done to prevent some TCP implementations from misbehaving upon receiving options in non-SYN segments. This is not a problem for this option, since the SYN,ACK sent during connection negotiation will not be signed and will thus be ignored. The connection will never be made, and non-SYN segments with options will never be sent. More importantly, the sending of authentication data must be under the complete control of the application, not at the mercy of the remote host not understanding the option.

5. Applications

The mechanisms described in this memo are intended for use with applications that manipulate key set contents. An application might toggle the active and eligible flags based upons system time, connection duration or any other external event.

6. TCP Enhanced Authentication Option

Figure 1 depicts the TCP Enhanced Authentication Option format.

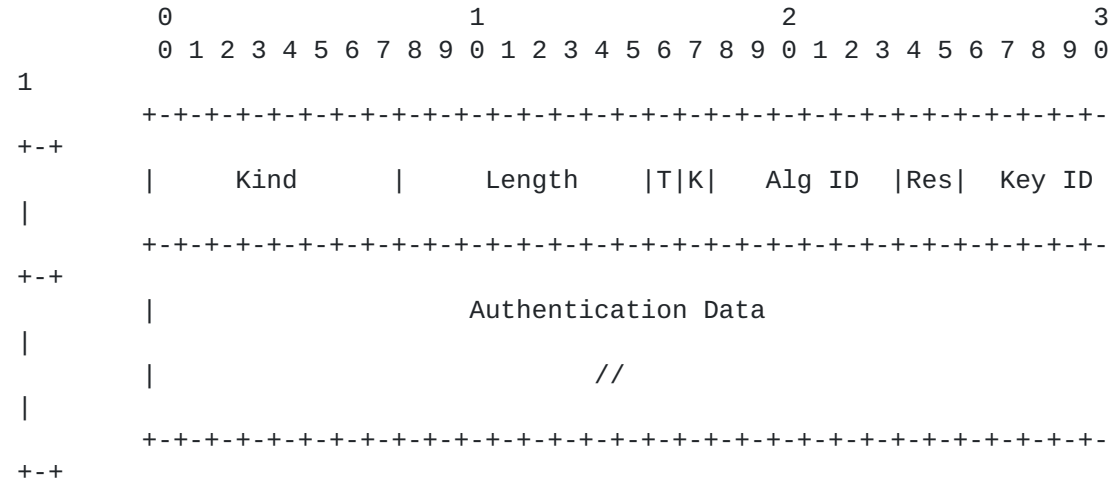

Figure 1: Option Syntax

Kind: 8 bits

The Kind field identifies the TCP Enhanced Authentication Option. This value will be assigned by IANA.

Length: 8 bits

The Length field specifies the length of the TCP Enhanced Authentication Option, in octets. This count includes two octets representing the Kind and Length fields.

The valid range for this field is from 4 to 40 octets, inclusive. For all algorithms specified in this memo the value will be 16 octets.

T-Bit: 1 bit

The T-bit specifies whether TCP Options were omitted from the TCP header for the purpose of MAC calculation. A value of 1 indicates that all TCP options other than the Extended Authentication Option were omitted. A value of 0 indicates that TCP options were included.
The default value is 0. (See Section 8 of this document for MAC calculation details.)

K-Bit: 1 bit

This bit is reserved for future enhancement. Its value MUST be equal
to zero. See Section 11 for details.

Alg ID: 6 bits

The Alg ID field identifies the MAC algorithm. See Section 9 for permissible values.

Res: 2 bits

These bits are reserved. They MUST be set to zero.

Key ID: 6 bits

The Key ID field identifies the key that was used to generate the message digest.

Authentication Data: Variable length

The Authentication Data field contains data that is used to authenticate the TCP segment. This data includes, but need not be restricted to, a MAC. The length and format of the Authentication Data Field can be derived from the Alg ID. See Section 9 for details.

7. Key Attributes

A key set is a set of keys, where each key is {L[i], S[i], A[i], E[i]}:

i	Key identifier, integer (0...63)
L[i]	Authentication algorithm to use with key[i].
S[i]	Shared secret to use with key[i].
A[i]	Active flag to use with key[i]
E[i]	Eligible flag to use with key[i]

For the purpose of this document, key[i] is defined as the key whose identifer is equal to i.

A list of values for L[i] is provided in Section 9 of this document. The format of S[i] depends upon L[i]. Also see Section 9 for details.

L[i] and S[i] MUST be configured symmetrically on TCP peers. That is, if key[i] is configured on two peer systems, L[i] and S[i] must be configured identically on each system.

For each key set, exactly one element MUST have A[i] set. Zero or more elements MAY have E[i] set.

In general, network operators should avoid reusing shared secrets. The degree to which an operator can reuse keys is defined by local security policy.

During the lifetime of a TCP connection, network operators may add and delete keys from the key set. However, the network operator must ensure that the active key is always configured on both TCP endpoints.

8. MAC Calculation

The sending TCP calculates a MAC by applying the authentication algorithm from the active key to the following items in the order that they are listed:

- the TCP pseudo-header

- the TCP header, assuming a checksum of zero. (See below for a discussion of TCP Options within the the TCP header.)

- the TCP segment data (if any)

For IPv4, the pseudo-header is described in RFC 793 [2]. It includes the 32-bit source IP address, the 32-bit destination IP address, the zero-extended protocol number (to form 16 bits), and the 16-bit segment length. Note that this includes use of IPv4 via IPv4-mapped IPv6 addresses, in which case the source and destination IP addresses are from the IPv4 portions of the IPv6 source and destination addresses, respectively.

For IPv6, the pseudo-header is described in RFC 2460 [3]. It includes the 128-bit source IPv6 address, the 128-bit destination IPv6 address, the zero-extended next header value (to form 32 bits), and the 32-bit segment length.

The header and pseudo-header are in network byte order.

By default, for the purpose of MAC calculation, the TCP header includes all TCP options, including the TCP Enhanced Authentication Option with its Authentication Data Field (i.e., MAC) set to zero. However, TCP implementations MAY omit all other TCP options from the MAC calculation; the TCP Enhanced Authentication Option itself must still be included in the calculation, as described above. When implementation do so, they MUST set the T-bit in the TCP Enhanced Authentication Option.

The receiving TCP calculates the MAC in a manner identical to the sending TCP. However, it MUST examine the T-bit from the incoming TCP Enhanced Authentication Option to determine whether incoming TCP Options should be included in the MAC calculation.

9. Authentication Algorithms

The following MAC Algorithms are suitable for use with this option.

- AES-128-CMAC-96. AES with a 128-bit key in the CMAC mode of operation [4] [5]. When this algorithm is used, implementations MUST specify a value of 1 (in binary, 000001) in the TCP Enhanced Authentication Option Alg ID field. Also, the Authentication Data field must contain exactly 96 bits representing the MAC, truncated to that length, with the high-order bit first. The AES-128-CMAC-96 algorithm MUST be implemented for an implementation to conform to this specification.

- HMAC-SHA-1-96. SHA-1 with a 160-bit key in the HMAC mode of operation [6]. When this algorithm is used, implementations MUST specify a value of 2 (in binary, 000010) in the TCP Enhanced Authentication Option ALG ID field. Also, the Authentication Data field must contain exactly 96 bits representing the MAC, truncated to that length, with the high-order bit first. This algorithm MAY be implemented for an implementation to conform to this specification.

The above algorithms are expected to safe to use in this application for many years. New algorithms may be added to this list as necessary, but it is important that they be properly vetted by the cryptographic community. To this end, the above algorithms are described in a list maintained by IANA, and the algorithm identifier associated with the algorithm is placed in the TCP Authentication Option header. Other algorithms may also be defined by an implementation using a Private Use identifier, but the suitability of those algorithms when used with the TCP Extended Authentication option is not assured.

Implementations MUST present a management interface through which the user can specify any member of the key space. For example, if the key contains 128 bits, the command line interface might accept this value as a string of exactly 32 hexadecimal digits, with each hexadecimal digit representing 4 bits of the shared secret.

Implementations MAY employ authentication algorithms not listed above.

10. Migration Issues

Either the TCP Enhanced Authentication Option or RFC 2385 may be applied to a TCP segment, but the two options SHOULD NOT be present in the same TCP segment. An implementation MAY support both options for a particular TCP session during migration from RFC 2385, but they MUST use different keys so as not to weaken the security of the TCP Enhanced Authentication Option (see the Security Considerations section for details). A receiver SHOULD accept either option. A sender MAY choose to continue sending RFC 2385 options until it has evidence that the other TCP endpoint shows use of the TCP Enhanced Authentication Option, in which case it migrates to the TCP Enhanced Authentication Option.

11. Future Enhancements

In the future, the TCP Enhanced Authentication Option will be

enhanced to support automated session key distribution. The K-bit is reserved for the purpose of indicating that session key distribution extensions are present. These extensions are beyond the scope of this memo.

12. Implications

12.1. Connectionless Resets

A connectionless reset will be ignored by the receiver of the reset if the originator of that reset does not know the key and therefore cannot generate the proper authentication data for the segment. This means, for example, that connection attempts by a TCP which is generating authentication data to a port with no listener will time out instead of being refused. Similarly, resets generated by a TCP in response to segments sent on a stale connection will also be ignored. Operationally this can be a problem since resets help some protocols recover quickly from peer crashes.

12.2. Performance

The performance hit in calculating digests may inhibit the use of this option. Performance will vary depending upon processor type, authentication algorithm, packet size and number of MAC calculations per second.

12.3. TCP Header Size

As with other options that are added to every segment, the size of the TCP Enhanced Authentication Option must be factored into the MSS offered to the other side during connection negotiation. Specifically, the size of the header to subtract from the MTU (whether it is the MTU of the outgoing interface or IP's minimal MTU of 576 octets) is now increased by the size of the TCP Enhanced Authentication Option.

The total header size is also an issue. The TCP header specifies where segment data starts with a 4-bit field which gives the total size of the header (including options) in 32-byte words. This means that the total size of the header plus options must be less than or equal to 60 octets. This leaves 40 octets for options.

As a concrete example, assume that an implementation defaults to sending window-scaling and timestamp information for connections it initiates. The most loaded segment will be the initial SYN packet to start the connection. With the TCP Enhanced Authentication Option using AES-128-CMAC-96, the SYN packet will contain the following:

-- 4 octets MSS option

-- 4 octets window scale option (3 octets padded to 4 in many implementations)

-- 12 octets for timestamp

-- 16 octets for the TCP Enhanced Authentication Option

This sums to 36 octets, leaving only four octets for future expansion.

12.4. Backwards Compatibility

On any particular TCP connection, use of the TCP Enhanced Authentication Option precludes use of the TCP MD5 Signature Option. However, use of the TCP Enhanced Authentication Option on one connection does not preclude the use of the TCP MD5 Signature Option on another connection by the same system.

12.5. ICMP-based attacks

The mechanism described in this document does not provide significant protection against ICMP-based attacks [13].

12.6. Relationship With TLS

The Transport Layer Security protocol (TLS) [14] provides confidentiality and message authentication to TCP connections. However, TLS works above the TCP layer, and does not protect the TCP connection itself. In contrast, the TCP Authentication Option provides protection against attacks on the TCP layer, such as connection reset attacks.

13. Contributors

The following individuals contributed to this document:

Chandrashekhar Appanna (achandra@cisco.com)

Andy Heffernan (ahh@juniper.net)

Kapil Jain (kapil@juniper.net)

David McGrew (mcgrew@cisco.com)

Satish Mynam (mynam@cisco.com)

Anantha Ramaiah (ananth@cisco.com)

14. Acknowledgments

Thanks to Steve Bellovin, Ted Faber, Ross Callon, Joe Touch and Ran Atkinson for their comments regarding this draft.

15. Security Considerations

This proposal describes a strong authentication method for authenticating TCP segments. It defines the use of cryptographic MAC algorithms, which are considered state-of-the-art. As such, their expected lifetime of usefulness extends for several years. But cryptographic algorithms have an effective lifetime, depending on advancing processor speed and cryptographic research. This proposal provides for the future addition of new MAC algorithms as they are needed.

Management of RFC 2385 keys has been a significant operational problem, both in terms of key synchronization and key selection. Current guidance [9] warns against sharing RFC 2385 keys between systems, and recommends changing keys according to a schedule. The same general operational issues are relevant for the management of MAC keys.

Because the TCP Authentication Option relies on manual configuration, it is possible that misconfigurations will occur. We review the scenarios and describe their impact on security.

When multiple devices are configured with the same key, it is possible that one or more of the devices is configured to use the wrong MAC algorithm. If the misconfigured device is using a MAC that is significantly weaker than that used by the correctly configured devices, where the weakness allows an attacker to recover the MAC key, the misconfiguration reduces the security of the properly configured devices. An attacker who can recover the key through cryptanalysis of the weaker algorithm can use that information to attack the stronger algorithm. For this reason, implementations SHOULD verify the length of the keys entered into the system, and reject keys that are too short. The extent of vulnerability will also be reduced when the receiver discards TCP segments due to a MAC Algorithm ID mismatch (i.e., the MAC Algorithm ID field in the TCP

segment and the MAC Algorithm ID associated with the key do not match). When this event is detected, it SHOULD provide that information to an administrator, e.g. through logging or a management interface. This attack is not applicable to AES-CMAC or HMAC, since neither of those MACs is vulnerable to a key recovery attack.

When one or more devices are configured with a particular key, it is possible that another device is configured with a slightly different key, due to a typographical error. For example, the two keys might differ only in a single hexadecimal digit. Message authentication codes that are vulnerable whenever two related keys are used could be vulnerable in this scenario. In order to protect against this potential vulnerability, it is RECOMMENDED that no MACs with such vulnerabilities be used. Neither AES-CMAC nor HMAC have such a vulnerability.

16. IANA Considerations

The terms "Standards Action" and "Private Use" in this section indicate the polices described for these terms in [7].

A new TCP Option Kind value must be defined in the IANA TCP Parameters registry.

The option header contains an 8-bit ALG ID, for which IANA is to create and maintain a registry entitled "MAC Algorithm IDs". This document defines the following message authentication code types:

MAC Algorithm ID	Value
RESERVED	0
AES-128-CMAC-96	1
HMAC-SHA-1-96	2
Standards Action	3-47
Private Use	48-63

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

17. References

17.1. Normative References

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

[2] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[3] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[4] National Institute of Standards and Technology, "Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication", FIPS PUB 800-38B, May 2005, <http://csrc.nist.gov/publications/nistpubs/800-38B/SP_800-38B.pdf>.

[5] Song, JH., Poovendran, R., Lee, J., and T. Iwata, "The AES-CMAC Algorithm", RFC 4493, June 2006.

[6] National Institute of Standards and Technology, "The Keyed-Hash Message Authentication Code (HMAC)", FIPS PUB 198, March 2002, <http://csrc.nist.gov/publications/fips/fips198/fips-198a.pdf>.

[7] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

17.2. Informative References

[8] Heffernan, A., "Protection of BGP Sessions via the TCP MD5 Signature Option", RFC 2385, August 1998.

[9] Leech, M., "Key Management Considerations for the TCP MD5 Signature Option", RFC 3562, July 2003.

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

[11] Bellare, M., Canetti, R., and H. Krawczyk, "Keying Hash Functions for Message Authentication", Proceedings of Crypto'96 , LNCS 1109, pp. 1-15., June 1996, <An extended version of this paper is available at http://www.research.ibm.com/security/bck2.ps>.

[12] Jacobson, V., Braden, B., and D. Borman, "TCP Extensions for High Performance", RFC 1323, May 1992.

[13] Gont, F., "ICMP attacks against TCP", draft-ietf-tcpm-icmp-attacks-01 (work in progress), October 2006.

[14] Dierks, T. and C. Allen, "The TLS Protocol Version 1.0",

RFC 2246, January 1999.

Authors' Addresses

Ronald P. Bonica
Juniper Networks
2251 Corporate Park Drive
Herndon, VA 20171
US

Email: rbonica@juniper.net

Brian Weis
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134-1706
US

Email: bew@cisco.com

Sriram Viswanathan
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134
US

Email: sriram_v@cisco.com

Andrew Lange
Alcatel
710 E. Middlefield Road
Mountain View, CA 94043
US

Email: andrew.lange@alcatel.com

Owen N. Wheeler
British Telecommunications plc
Adastral Park
Martlesham Heath
IPSWICH, Suffolk IP5 3RE
GB

Email: owen.wheeler@bt.com

Full Copyright Statement

Intellectual Property

Acknowledgment

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