TCPM WG
Internet Draft
Intended status: Standards Track
Expires: April 2017

J. Touch
USC/ISI

October 21, 2016

A TCP Authentication Option Extension for Payload Encryption
draft-touch-tcp-ao-encrypt-06.txt

Status of this Memo

Copyright Notice

Abstract

This document describes an extension to the TCP Authentication Option (TCP-AO) to encrypt the TCP segment payload in addition to providing TCP-AO's authentication of the payload, TCP header, and IP pseudoheader. This extension augments how the packet contents and headers are processed and which keys are derived, and adds a capability for in-band coordination of unauthenticated Diffie-Hellman key exchange at connection establishment. The extension preserves key rollover coordination and protection of long-lived connections.

Table of Contents

1. Introduction

This document describes an extension to the TCP Authentication Option (TCP-AO) [RFC5925] called TCP-AO-ENC to support its use to encrypt TCP segment payload contents in addition to authenticating the segment. TCP-AO-ENC is intended for use where TCP user data privacy is required and where TCP control protocol protection is also needed.

TCP-AO-ENC supports two different modes: authenticated encryption and unauthenticated (BTNS-style) encryption [RFC5387]. Authenticated mode (ENC-AUTH) relies on out-of-band coordination of master key tuples in the same way as TCP-AO, and protects all segments of a connection. Unauthenticated (ENC-BTNS) mode supports in-band unsigned Diffie-Hellman key exchange during the initial SYN, and protects connections from all except man-in-the-middle attacks during connection establishment.

This document assumes detailed familiarity with TCP-AO [RFC5925]. TCP-AO-ENC extends how TCP-AO generates traffic keys and how those keys are used to process TCP segment headers and payloads, but does not otherwise alter other aspects of the TCP-AO mechanism [RFC5926].

2. 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 RFC-2119 [RFC2119]. When used in lower case, these words have their conventional meaning and do not convey the interpretations in RFC-2119.

3. Background

The premise of TCP-AO-ENC is that it might be useful to allow TCP-AO to encrypt TCP segment payloads, in addition to authenticating the entire segment.

This is accomplished by the following additions, as a preview:

- An encryption flag to indicate when segment payload encryption is used.

- Traffic encryption key, in addition to the TCP-AO traffic (authentication) key. TCP-AO-ENC can be used with only symmetric ciphers that avoiding the need for padding (stream ciphers).

- Augment input and output processing to include encryption/decryption.

TCP-AO-ENC does not change any other aspects of TCP-AO [RFC5925], and is compatible with TCP-AO-NAT [RFC6978]. TCP-AO-NAT is intended for use only where coordinated between endpoints for connections that match the shared Master Key Tuple (MKT) parameters, as with all other MKT parameters.

4. Authenticated and Unauthenticated Modes

TCP-AO-ENC includes two modes: authenticated (ENC-AUTH) and unauthenticated (ENC-BTNS). The latter is consistent with the "Better Than Nothing Security" approach to protect communication in the absence of public key infrastructure (PKI) or pre-shared keys [RFC5387].

ENC-AUTH mode operates in the same way as original TCP-AO authentication, by relying on out-of-band Master Key Tuples (MKTs) that are deployed in advance of new connections. All segments of connections covered by ENC-AUTH are encrypted and authenticated using keying material derived from those MKTs.

ENC-BTNS mode can be used without an out-of-band key exchange protocol. It exchanges unauthenticated, unsigned Diffie-Hellman nonces during connection establishment in-band, and uses that information to derive keys used to protect the remainder of the connection. ENC-BTNS mode is susceptible to man-in-the-middle attacks in which the adversary both participates in the initial nonce exchange and processes subsequent segments; this protection increases the effort of the attacker and can help avoid low-effort DDOS attacks that disrupt established connections [RFC4953].

Because ENC-BTNS uses in-band nonce exchange only during the initial SYN, TCP-AO key rollover is not used in that mode. The KeyIDs used during the nonce exchange are recorded and used throughout the connection.

5. Extension for Payload Encryption

The following describe the additions to TCP-AO needed to support TCP-AO-ENC.

5.1. Additional Master Key Tuple components

TCP-AO-ENC augments the MKT as follows; as with other MKT components, these MUST NOT change during a connection:

- TCP encryption mode. This indicates the use and mode (ENC-AUTH or ENC-BTNS) of segment payload encryption, or is clear when encryption is not used (e.g., for conventional TCP-AO).

- Encryption Key Derivation Function (E-KDF). Indicates the key derivation function and its parameters, as used to generate traffic encryption keys from master keys in the same way that the TCP-AO KDG generates traffic (authentication) keys.

- Encryption algorithm. Indicates the encryption algorithm and its parameters as used for encrypted connections.

PTCP-AO-ENC processes TCP packets in the same way as TCP-AO, except that it replaces the authentication input and output processing as follows:

5.2. Additional traffic keys

TCP-AO-ENC uses the E-KDF to derive four additional keys used for traffic encryption:

- Send_SYN_traffic_encryption_key
- Receive_SYN_traffic_encryption_key
- Send_other_traffic_encryption_key
- Receive_other_traffic_encryption_key

5.3. Per-Connection TCP-AO Parameters

The per-connection TCP-AO parameters are not affected by the use of TCP-AO-ENC-AUTH, except that MKTs indicated by Current_key and Rnext_key would indicate the use of payload encryption.

The per-connection TCP-AO parameters for TCP-AO-ENC-BTNS are augmented by the addition of the following Diffie-Hellman nonces:

- Send_nonce. The locally-generated Diffie-Hellman nonce.
- Receive_nonce. The Diffie-Hellman nonce generated by the remote end of the connection.

These nonces are exchanged during the initial SYN exchange in ENC-BTNS mode; for ENC-AUTH mode, similar information is exchanged out-of-band and is used to derive the encryption keys. KeyIDs used with these nonces are recorded during nonce exchange and used for the remainder of the connection.

The use of payload encryption as specified in these MKTs SHOULD NOT change during a TCP connection.

5.4. Traffic Encryption Key Derivation Functions

Traffic encryption keys are derived from the MKTs using the E-KDF, in the same way and used on the same segments as their corresponding authentication keys, e.g.:

- Send_SYN_traffic_encryption_key / Send_SYN_traffic_key
- Receive_SYN_traffic_encryption_key / Receive_SYN_traffic__key
- Send_other_traffic_encryption_key / Send_other_traffic__key
- Receive_other_traffic_encryption_key / Receive_other_traffic_key

6. TCP-AO-ENC Interaction with TCP

TCP-AO-ENC augments TCP segment send and receive processing to include encryption/decryption. Note that the encryption initialization vector MAY depend on TCP header state, but MUST NOT depend on the processing of previous segments because segments may arrive (and need to be decrypted) out of order.

6.1. Sending TCP Segments

For ENC-BTNS, initial SYN and SYN-ACK are used to establish the Diffie-Hellman nonces as follows:

- Initial SYN and SYN-ACK. The initial SYN (SYN and not ACK) and SYN-ACK segments are not encrypted or authenticated. Instead, their HMAC field contains the Diffie-Hellman nonce in network-standard byte order.

The size of the Diffie-Hellman nonce determines the strength of the resulting association. For TCP, such nonces are limited to the available option space in the SYN and SYN-ACK segments. This currently limits the nonce to 128 bits (16 bytes). Larger nonces can be supported using extensions to expand the TCP SYN option space [Bo14][Br14][To16].

Because these segments are not authenticated or encrypted, they SHOULD NOT contain user data. In a typical client-server system, user data usually commences in other segments anyway.

All other TCP segments are processed as follows:

1. The segment payload is encrypted in-place using the traffic encryption key.

2. The segment is authenticated using TCP-AO as per [RFC5925].

6.2. Receiving TCP Segments

For ENC-BTNS, initial SYN and SYN-ACK are used to establish the Diffie-Hellman nonces as follows:

- Initial SYN and SYN-ACK. The unauthenticated Diffie-Hellman nonce is extracted from the HMAC field, and used to construct the encryption and traffic keys for the connection. Because these segments are not encrypted or authenticated, no further processing is required.

All other incoming TCP segments are processed as follows:

1. TCP-AO authenticates the segment, including discarding it if authentication fails, as per [RFC5925].

2. The segment payload is decrypted in-place using the traffic encryption key.

6.3. Other TCP Impact

TCP-AO-ENC has no impact on TCP beyond that of TCP-AO, including impact on TCP header size, connectionless resets, and ICMP handling.

TCP-AO-ENC is compatible with the use of TCP-AO-NAT if traversal of NAT boxes is desired.

7. Security Considerations

TCP-AO-ENC augments TCP-AO to provide segment payload privacy.

TCP-AO-ENC relies on TCP-AO's authentication to avoid replay attacks and to ensure that the segments originate from the intended source.

TCP-AO-ENC supports only stream ciphers because the TCP segment must be encrypted and decrypted in-situ. Support for padding would require additional option space to indicate the original message length, and this complication does not seem necessary.

The design of TCP-AO-ENC can support either symmetric or asymmetric keys. However, because TCP-AO derives traffic (authentication) keys from MKTs using KDFs, it was deemed sufficient that TCP-AO-ENC derive traffic encryption keys from MKTs using E-KDFs in a similar manner, and both endpoints would thus derive the same traffic encryption keys just as they derive the same traffic

(authentication) keys. Extensions of TCP-AO-ENC to support asymmetric keying are possible if traffic keys are managed using an out-of-band mechanism, but not if they are derived from MKTs.

ENC-AUTH has no additional security considerations. ENC-BTNS cannot authenticate or encrypt the segments used for nonce exchange, i.e., the initial SYN and SYN-ACK. As a result, ENC-BTNS is susceptible to man-in-the-middle attacks during connection establishment, but remains useful to ensure that established connections are protected.

8. Keying Algorithms

TCP-AO-ENC algorithms are specified in a separate document, as was the custom for TCP-AO.

NOTES [for that doc]:

- E-KDF - also, can a MKT use the same alg for KDF and E-KDF?

- Encryption algorithm - possibilities include AES CTR (CTR initial value can be the ESN) or AES CBC and Camellia CBC as per TLS 1.2.

9. IANA Considerations

There are no IANA considerations for this document. This section can be removed upon publication as an RFC.

10. References

10.1. Normative References

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

[RFC5925] Touch, J., A. Mankin, R. Bonica, "The TCP Authentication Option", RFC 5925, Jun. 2010.

[RFC5926] Lebovitz, G., E. Rescorla, "Cryptographic Algorithms for the TCP Authentication Option (TCP-AO)", RFC 5926, June 2010.

10.2. Informative References

[Bo14] Borman, D., "TCP Four-Way Handshake", draft-borman-tcp4way-00 (work in progress), Oct. 2014.

[Br14] Briscoe, B., "Inner Space for TCP Options", draft-briscoe-tcpm-inner-space-01 (work in progress), Oct. 2014.

[RFC4953] J. Touch, "Defending TCP Against Spoofing Attacks", RFC 4953, July 2007.

[RFC5387] Touch, J., Black, D., and Y. Wang, "Problem and Applicability Statement for Better-Than-Nothing Security (BTNS)", RFC 5387, November 2008.

[RFC6978] Touch, J., "A TCP Authentication Option Extension for NAT Traversal", RFC 6978, July 2013.

[To16] Touch, J., and T. Faber, "TCP SYN Extended Option Space Using an Out-of-Band Segment", draft-touch-tcpm-tcp-syn-ext-opt-05 (work in progress), Oct. 2016.

11. Acknowledgments

This extension was informed by discussions with Gene Tsudik, and various members of the TCPM and TCPCRYPT mailing lists, including Christian von Roques.

This work is partly supported by USC/ISI's Postel Center.

This document was prepared using 2-Word-v2.0.template.dot.

Author's Address

Joe Touch
USC/ISI
4676 Admiralty Way
Marina del Rey, CA 90292
USA

Phone: +1 (310) 448-9151
Email: touch@isi.edu