TCPM WG
Internet Draft
Intended status: Standards Track
Expires: September 2014

J. Touch
USC/ISI

March 19, 2014

A TCP Authentication Option Extension for Payload Encryption
draft-touch-tcp-ao-encrypt-00.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 changes how the packet contents and headers are processed and which keys are derived, but not its key coordination or 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.

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 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 MKT parameters, as with all other MKT parameters.

4. Extension for Payload Encryption

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

4.1. 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 flag. When present, this indicates the use of segment payload encryption.

- 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:

4.2. 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

4.3. Per-Connection TCP-AO Parameters

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

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

4.4. 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

5. 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.

5.1. Sending TCP Segments

Outgoing 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].

5.2. Receiving TCP Segments

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.

5.3. 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.

6. 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.

7. To be completed...

Where are required algorithms specified? This doc or a separate one?

- 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.

8. IANA Considerations

(TO BE CONFIRMED, BUT FOR NOW:)

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

9. References

9.1. 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.

9.2. Informative References

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

10. Acknowledgments

This extension was informed by discussions with Gene Tsudik.

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