              Datagram Transport Layer Security (DTLS)
          for Stream Control Transmission Protocol (SCTP)

Abstract

   This document describes the usage of the Datagram Transport Layer
   Security (DTLS) protocol over the Stream Control Transmission
   Protocol (SCTP).

   DTLS over SCTP provides communications privacy for applications that
   use SCTP as their transport protocol and allows client/server
   applications to communicate in a way that is designed to prevent
   eavesdropping and detect tampering or message forgery.

   Applications using DTLS over SCTP can use almost all transport
   features provided by SCTP and its extensions.

Status of This Memo

   This is an Internet Standards Track document.

   This document is a product of the Internet Engineering Task Force
   (IETF). It represents the consensus of the IETF community. It has
   received public review and has been approved for publication by the
   Internet Engineering Steering Group (IESG). Further information on
   Internet Standards is available in Section 2 of RFC 5741.

   Information about the current status of this document, any errata,
   and how to provide feedback on it may be obtained at
   http://www.rfc-editor.org/info/rfc6083.

Copyright Notice

Table of Contents

1. Introduction

1.1. Overview

   This document describes the usage of the Datagram Transport Layer
   Security (DTLS) protocol, as defined in [RFC4347], over the Stream
   Control Transmission Protocol (SCTP), as defined in [RFC4960].

   DTLS over SCTP provides communications privacy for applications that
   use SCTP as their transport protocol and allows client/server
   applications to communicate in a way that is designed to prevent
   eavesdropping and detect tampering or message forgery.

   Applications using DTLS over SCTP can use almost all transport
   features provided by SCTP and its extensions.

   TLS, from which DTLS was derived, is designed to run on top of a
   byte-stream-oriented transport protocol providing a reliable, in-
   sequence delivery. Thus, TLS is currently mainly being used on top
   of the Transmission Control Protocol (TCP), as defined in [RFC0793].

TLS over SCTP as described in [RFC3436] has some serious limitations:

o It does not support the unordered delivery of SCTP user messages.

o It does not support partial reliability as defined in [RFC3758].

o It only supports the usage of the same number of streams in both
   directions.

o It uses a TLS connection for every bidirectional stream, which
   requires a substantial amount of resources and message exchanges
   if a large number of streams is used.

DTLS over SCTP as described in this document overcomes these
limitations of TLS over SCTP. In particular, DTLS/SCTP supports:

o preservation of message boundaries.

o a large number of unidirectional and bidirectional streams.

o ordered and unordered delivery of SCTP user messages.

o the partial reliability extension as defined in [RFC3758].

o the dynamic address reconfiguration extension as defined in
   [RFC5061].

However, the following limitations still apply:

o The maximum user message size is 2^14 bytes, which is the DTLS
   limit.

o The DTLS user cannot perform the SCTP-AUTH key management because
   this is done by the DTLS layer.

The method described in this document requires that the SCTP
implementation supports the optional feature of fragmentation of SCTP
user messages as defined in [RFC4960] and the SCTP authentication
extension defined in [RFC4895].

1.2. Terminology

This document uses the following terms:

Association: An SCTP association.

Stream: A unidirectional stream of an SCTP association. It is
   uniquely identified by a stream identifier.

1.3. Abbreviations

   DTLS: Datagram Transport Layer Security

   MTU: Maximum Transmission Unit

   PPID: Payload Protocol Identifier

   SCTP: Stream Control Transmission Protocol

   TCP: Transmission Control Protocol

   TLS: Transport Layer Security

2. Conventions

   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].

3. DTLS Considerations

3.1. Version of DTLS

   This document is based on [RFC4347], and it is expected that DTLS/
   SCTP as described in this document will work with future versions of
   DTLS.

3.2. Message Sizes

   DTLS limits the DTLS user message size to the current Path MTU minus
   the header sizes. For the purposes of running over SCTP, the DTLS
   path MTU MUST be considered to be 2^14.

3.3. Replay Detection

   The replay detection of DTLS may result in the DTLS layer dropping
   messages. Since DTLS/SCTP provides a reliable service if requested
   by the application, replay detection cannot be used. Therefore,
   replay detection of DTLS MUST NOT be used.

3.4. Path MTU Discovery

   SCTP provides Path MTU discovery and fragmentation/reassembly for
   user messages. According to Section 3.2, DTLS can send maximum sized
   messages. Therefore, Path MTU discovery of DTLS MUST NOT be used.

3.5. Retransmission of Messages

   SCTP provides a reliable and in-sequence transport service for DTLS
   messages that require it. See Section 4.4. Therefore, DTLS
   procedures for retransmissions MUST NOT be used.

4. SCTP Considerations

4.1. Mapping of DTLS Records

   The supported maximum length of SCTP user messages MUST be at least
   2^14 + 2048 + 13 = 18445 bytes (2^14 + 2048 is the maximum length of
   the DTLSCiphertext.fragment, and 13 is the size of the DTLS record
   header). In particular, the SCTP implementation MUST support
   fragmentation of user messages.

   Every SCTP user message MUST consist of exactly one DTLS record.

4.2. DTLS Connection Handling

   Each DTLS connection MUST be established and terminated within the
   same SCTP association. A DTLS connection MUST NOT span multiple SCTP
   associations.

4.3. Payload Protocol Identifier Usage

   Application protocols using DTLS over SCTP SHOULD register and use a
   separate payload protocol identifier (PPID) and SHOULD NOT reuse the
   PPID that they registered for running directly over SCTP.

   Using the same PPID does not harm as long as the application can
   determine whether or not DTLS is used. However, for protocol
   analyzers, for example, it is much easier if a separate PPID is used.

   This means, in particular, that there is no specific PPID for DTLS.

4.4. Stream Usage

   All DTLS messages of the ChangeCipherSpec, Alert, or Handshake
   protocol MUST be transported on stream 0 with unlimited reliability
   and with the ordered delivery feature.

   DTLS messages of the ApplicationData protocol SHOULD use multiple
   streams other than stream 0; they MAY use stream 0 for everything if
   they do not care about minimizing head of line blocking.

4.5. Chunk Handling

   DATA chunks of SCTP MUST be sent in an authenticated way as described
   in [RFC4895]. Other chunks MAY be sent in an authenticated way.
   This makes sure that an attacker cannot modify the stream in which a
   message is sent or affect the ordered/unordered delivery of the
   message.

   If PR-SCTP as defined in [RFC3758] is used, FORWARD-TSN chunks MUST
   also be sent in an authenticated way as described in [RFC4895]. This
   makes sure that it is not possible for an attacker to drop messages
   and use forged FORWARD-TSN, SACK, and/or SHUTDOWN chunks to hide this
   dropping.

4.6. Renegotiation

   DTLS supports renegotiation, and therefore this feature is also
   available by DTLS/SCTP. It is up to the upper layer to use/allow it
   or not. Application writers should be aware that allowing
   renegotiations may result in changes of security parameters.

4.7. Handshake

   A DTLS implementation discards DTLS messages from older epochs after
   some time, as described in Section 4.1 of [RFC4347]. This is not
   acceptable when the DTLS user performs a reliable data transfer. To
   avoid discarding messages, the following procedures are required.

   Before sending a ChangeCipherSpec message, all outstanding SCTP user
   messages MUST have been acknowledged by the SCTP peer and MUST NOT be
   revoked by the SCTP peer.

   Prior to processing a received ChangeCipherSpec, all other received
   SCTP user messages that are buffered in the SCTP layer MUST be read
   and processed by DTLS.

   User messages that arrive between ChangeCipherSpec and Finished
   messages and use the new epoch have probably passed the Finished
   message and MUST be buffered by DTLS until the Finished message is
   read.

4.8. Handling of Endpoint-Pair Shared Secrets

   The endpoint-pair shared secret for Shared Key Identifier 0 is empty
   and MUST be used when establishing a DTLS connection. Whenever the
   master key changes, a 64-byte shared secret is derived from every
   master secret and provided as a new endpoint-pair shared secret by
   using the exporter described in [RFC5705]. The exporter MUST use the

label given in Section 5 and no context. The new Shared Key
Identifier MUST be the old Shared Key Identifier incremented by 1.
If the old one is 65535, the new one MUST be 1.

Before sending the Finished message, the active SCTP-AUTH key MUST be
switched to the new one.

Once the corresponding Finished message from the peer has been
received, the old SCTP-AUTH key SHOULD be removed.

4.9. Shutdown

To prevent DTLS from discarding DTLS user messages while it is
shutting down, a CloseNotify message MUST only be sent after all
outstanding SCTP user messages have been acknowledged by the SCTP
peer and MUST NOT still be revoked by the SCTP peer.

Prior to processing a received CloseNotify, all other received SCTP
user messages that are buffered in the SCTP layer MUST be read and
processed by DTLS.

5. IANA Considerations

IANA added a value to the TLS Exporter Label registry as described in
[RFC5705]. The label is "EXPORTER_DTLS_OVER_SCTP".

6. Security Considerations

The security considerations given in [RFC4347], [RFC4895], and
[RFC4960] also apply to this document.

It is possible to authenticate DTLS endpoints based on IP addresses
in certificates. SCTP associations can use multiple addresses per
SCTP endpoint. Therefore, it is possible that DTLS records will be
sent from a different IP address than that originally authenticated.
This is not a problem provided that no security decisions are made
based on that IP address. This is a special case of a general rule:
all decisions should be based on the peer's authenticated identity,
not on its transport layer identity.

For each message, the SCTP user also provides a stream identifier, a
flag to indicate whether the message is sent ordered or unordered,
and a payload protocol identifier. Although DTLS can be used to
provide privacy for the actual user message, none of these three are
protected by DTLS. They are sent as clear text, because they are
part of the SCTP DATA chunk header.

DTLS supports cipher suites that contain a NULL cipher algorithm.
Negotiating a NULL cipher algorithm will not provide communications
privacy for applications and will not provide privacy for user
messages.

7. Acknowledgments

   The authors wish to thank Anna Brunstrom, Lars Eggert, Gorry
   Fairhurst, Ian Goldberg, Alfred Hoenes, Carsten Hohendorf, Stefan
   Lindskog, Daniel Mentz, and Sean Turner for their invaluable
   comments.

8. References

8.1. Normative References

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

   [RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P.
              Conrad, "Stream Control Transmission Protocol (SCTP)
              Partial Reliability Extension", RFC 3758, May 2004.

   [RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer
              Security", RFC 4347, April 2006.

   [RFC4895] Tuexen, M., Stewart, R., Lei, P., and E. Rescorla,
              "Authenticated Chunks for the Stream Control Transmission
              Protocol (SCTP)", RFC 4895, August 2007.

   [RFC4960] Stewart, R., "Stream Control Transmission Protocol",
              RFC 4960, September 2007.

   [RFC5705] Rescorla, E., "Keying Material Exporters for Transport
              Layer Security (TLS)", RFC 5705, March 2010.

8.2. Informative References

   [RFC0793] Postel, J., "Transmission Control Protocol", STD 7,
              RFC 793, September 1981.

   [RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport
              Layer Security over Stream Control Transmission Protocol",
              RFC 3436, December 2002.

   [RFC5061]   Stewart, R., Xie, Q., Tuexen, M., Maruyama, S., and M.
               Kozuka, "Stream Control Transmission Protocol (SCTP)
               Dynamic Address Reconfiguration", RFC 5061,
               September 2007.

Authors' Addresses

   Michael Tuexen
   Muenster University of Applied Sciences
   Stegerwaldstr. 39
   48565 Steinfurt
   Germany

   EMail: tuexen@fh-muenster.de

   Robin Seggelmann
   Muenster University of Applied Sciences
   Stegerwaldstr. 39
   48565 Steinfurt
   Germany

   EMail: seggelmann@fh-muenster.de

   Eric Rescorla
   RTFM, Inc.
   2064 Edgewood Drive
   Palo Alto, CA 94303
   USA

   EMail: ekr@networkresonance.com