Network Working Group R. Seggelmann
Internet-Draft M. Tuexen
Intended status: Standards Track Muenster Univ. of Applied Sciences
Expires: February 22, 2010 M. Williams
August 21, 2009

Transport Layer Security and Datagram Transport Layer Security Heartbeat Extension

draft-seggelmann-tls-dtls-heartbeat-01.txt

Status of this Memo

Copyright Notice

Abstract

This document describes the Heartbeat Extension for the Transport

Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocol.

The Heartbeat Extension provides a new protocol for TLS/DTLS allowing the usage of keep-alive functionality without performing a renegotiation and a basis for PMTU discovery for DTLS.

Table of Contents

1. Introduction 3
2. Heartbeat Hello Extension 3
3. Heartbeat Protocol 4
4. Heartbeat Request and Response Messages 5
5. IANA Considerations 5
6. Security Considerations 5
7. Acknowledgments 5
8. References 5
8.1. Normative References 5
8.2. Informative References 6
Authors' Addresses 6

1. Introduction

1.1. Overview

This document describes the Heartbeat Extension for the Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) protocols, as defined in [RFC5246] and [RFC4347].

DTLS is designed to secure traffic running on top of unreliable transport protocols. Such protocols have usually no session management. The only mechanism available at the DTLS layer to figure out if a peer is still alive is performing a costly renegotiation. If the application uses unidirectional traffic there is no other way. Furthermore, DTLS needs to perform PMTU discovery but has no specific message type to realize it without affecting user message transfer.

TLS is based on reliable protocols but there is not necessarily a feature available to keep the connection alive without continuous data transfer.

The Heartbeat Extension as described in this document overcomes these limitations. The user can use the new HeartbeatRequest message which has to be answered by the peer with a HeartbeartResponse immediately. To perform PMTU discovery HeartbeatRequest messages containing padding can be used as described in [RFC4821] for SCTP using the PAD-chunk defined in [RFC4820].

1.2. 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].

2. Heartbeat Hello Extension

The support of Heartbeats is indicated with Hello Extensions. A peer can not only indicate that its implementation supports Heartbeats, it can also choose whether it is willed to receive and respond or only to send them. This decision can be changed with every renegotiation. HeartbeatRequests MUST NOT be sent to a peer denying acceptance.

```
   enum {
      peer_allowed_to_send(1),
      peer_not_allowed_to_send(2),
      (255)
   } HeartbeatMode;

   struct {
      HeartbeatMode mode;
   } HeartbeatExtension;
```

3. Heartbeat Protocol

The Heartbeat protocol is a new protocol on top of the Record Layer. The protocol itself consists of two message types, HeartbeatRequest and HeartbeatResponse.

```
   enum {
      heartbeat_request(1),
      heartbeat_response(2),
      (255)
   } HeartbeatMessageType;
```

Like the ChangeCipherSpec, a HeartbeatRequest can arrive at any time during the lifetime of a connection. Whenever a HeartbeatRequest is received, it has to be answered with a corresponding HeartbeatResponse message immediately.

However, a HeartbeatRequest message SHOULD NOT be sent during handshakes and there MUST NOT be more than one HeartbeatRequest message in flight at a time.

When using DTLS, HeartbeatRequest messages MUST be retransmitted using the simple timeout and retransmission scheme DTLS uses for flights. In particular, after a number of retransmissions without receiving a corresponding HeartbeatResponse message having the expected payload the DTLS connection SHOULD be terminated. The threshold used for this SHOULD be the same as for DTLS handshake messages.

When using TLS, HeartbeatRequest messages only need to be sent once. The transport layer will handle retransmissions. If no corresponding HeartbeatResponse message has been received after a user configured amount of time, the TLS connection SHOULD be terminated.

4. Heartbeat Request and Response Messages

The Heartbeat protocol messages consist of their type and an arbitrary payload and padding.

```
struct {
   HeartbeatMessageType type;
   opaque payload<0..2^14-5>;
   opaque padding<0..2^14-5>;
} HeartbeatMessage;
```

The length of payload and padding in total MUST NOT exceed 2^14-5 bytes.

When a HeartbeatRequest message is received, a corresponding HeartbeatResponse message MUST be sent carrying an exact copy of the payload of the HeartbeatRequest. The padding MUST NOT be included in the HeartbeatResponse message, i.e. the padding field has a length of zero.

If a received HeartbeatResponse message does not contain the expected payload the message MUST be discarded silently. If it does contain the expected payload the retransmission timer MUST be stopped.

5. IANA Considerations

The extension, content and message types have to be assigned by IANA.

6. Security Considerations

This document does not add any additional security considerations in addition to the ones given in [RFC4347] and [RFC5246].

7. Acknowledgments

The authors wish to thank Eric Rescorla, and Pasi Eronen 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.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

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

8.2. Informative References

[RFC4820] Tuexen, M., Stewart, R., and P. Lei, "Padding Chunk and Parameter for the Stream Control Transmission Protocol (SCTP)", RFC 4820, March 2007.

[RFC4821] Mathis, M. and J. Heffner, "Packetization Layer Path MTU Discovery", RFC 4821, March 2007.

Authors' Addresses

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

Email: seggelmann@fh-muenster.de

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

Email: tuexen@fh-muenster.de

Michael Williams

Email: michael.glenn.williams@gmail.com