Network Working Group R. Seggelmann
Internet-Draft M. Tuexen
Intended status: Standards Track Muenster Univ. of Appl. Sciences
Expires: January 12, 2012 M. Williams
July 11, 2011

Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension
draft-ietf-tls-dtls-heartbeat-02.txt

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 path maximum transmission unit (PMTU) discovery for DTLS.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 12, 2012.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

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] and their adoptions to specific transport protocol as described in [RFC3436], [RFC5238], and [RFC6083].

DTLS is designed to secure traffic running on top of unreliable transport protocols. Usually such protocols have 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 path maximum transmission unit (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 probe packets as described in [RFC4821].

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 willing to receive HeartbeatRequest messages and respond with HeartbeatResponse messages or only to send HeartbeatRequest messages. The former is indicated by using peer_allowed_to_send as the HeartbeatMode, the latter is indicated by using peer_not_allowed_to_send as the Heartbeat mode. This decision can be changed with every renegotiation. HeartbeatRequest messages MUST NOT be sent to a peer indicating peer_not_allowed_to_send. If an endpoint has indicated peer_not_allowed_to_send and receives a

HeartbeatRequest message SHOULD drop the message silently and MAY send an unexpected_message Alert message.

The format of the Heartbeat Hello Extension is defined by:

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

struct {
   HeartbeatMode mode;
} HeartbeatExtension;
```

Upon reception of an unknown mode, an error Alert message using illegal_parameter as its AlertDescription MUST be sent in response.

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

However, a HeartbeatRequest message SHOULD NOT be sent during handshakes. If a handshake is initiated while a HeartbeatRequest is still in flight, the sending peer MUST stop the retransmission timer for it. The receiving peer SHOULD discard it silently, if it arrives during or after the handshake. HeartbeatRequest messages from older epochs SHOULD be discarded.

There MUST NOT be more than one HeartbeatRequest message in flight at a time. A HeartbeatRequest message is considered to be in flight until the corresponding HeartbeatResponse message is received, or until the retransmit timer expires.

When using an unreliable transport protocol like DCCP or UDP, HeartbeatRequest messages MUST be retransmitted using the simple timeout and retransmission scheme DTLS uses for flights as described in Section 4.2.4 of [RFC4347]. 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. Please note, that after the timer supervising a HeartbeatRequest messages expires, this message is no longer considered in flight. Therefore the HeartbeatRequest message is eligible for retransmission. The retransmission scheme in combination with the restriction that only one HeartbeatRequest is allowed to be in flight ensures that the congestion control is handled appropriately in case of the transport protocol not providing one, like in the case of DTLS over UDP.

When using a reliable transport protocol like SCTP or TCP, 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 DTLS/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;
   uint16 payload_length;
   opaque payload[HeartbeatMessage.payload_length];
   opaque padding[padding_length];
} HeartbeatMessage;
```

The length of a HeartbeatMessage in total MUST NOT exceed 2^14 or max_fragment_length when negotiated as defined in [RFC6066].

type: The message type, either heartbeat_request or heartbeat_response.

payload_length: The length of the payload.

payload: The payload consists of arbitrary content.

padding: The padding is additional arbitrary content which MUST be ignored by the receiver. The padding_length is TLSPlaintext.length - payload_length - 3 with TLS and DTLSPlaintext.length - payload_length - 3 with DTLS.

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 of the received HeartbeatRequest message MUST be ignored. It MUST NOT be included in the HeartbeatResponse message, i.e. the padding field of the HeartbeatResponse message MUST have 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.

If payload_length is either shorter than expected and thus indicates padding in a HeartbeatResponse or exceeds the actual message length in any message type, an error Alert message using illegal_parameter as its AlertDescription MUST be sent in response.

5. Use Cases

5.1. Path MTU Discovery

DTLS performs path MTU discovery as described in Section 4.1.1.1 of [RFC4347]. A detailed description how to perform path MTU discovery is given in [RFC4821]. The necessary probe packets are the HeartbeatRequest messages.

This method using HeartbeatRequest messages for DTLS is similar to the one for the Stream Control Transmission Protocol (SCTP) using the padding chunk (PAD-chunk) defined in [RFC4820].

5.2. Liveliness check

Sending HeartbeatRequest messages allows the sender to make sure that it can reach the peer and the peer is alive. Even in case of TLS/TCP this allows this check at a much higher rate than the TCP keepalive feature would allow.

Besides making sure that the peer is still reachable, sending HeartbeatRequest messages refreshes the NAT state of all involved NATs.

HeartbeatRequest messages SHOULD only be sent after an idle period that is at least multiple round trip times long.

6. IANA Considerations

[NOTE to RFC-Editor:

"RFCXXXX" is to be replaced by the RFC number you assign this document.

]

IANA needs to assign the heartbeat content type (value TBD) from the TLS ContentType Registry as specified in [RFC5246]. The reference should be RFCXXXX.

IANA needs to maintain a new registry for Heartbeat Message Types. The message types are numbers in the range from 0 to 255 (decimal). Initially IANA needs to assign the heartbeat_request (suggested value 1) and the heartbeat_response (suggested value 2) message type. The values 0 and 255 should be reserved. This registry uses the Specification Required policy as described in [RFC5226]. The reference should be RFCXXXX.

IANA needs to assign the heartbeat extension type (value TBD) from the TLS Extension Type Registry as specified in [RFC5246]. The reference should be RFCXXXX.

IANA needs to maintain a new registry for Heartbeat Modes. The modes are numbers in the range from 0 to 255 (decimal). Initially IANA needs to assign the peer_allowed_to_send (suggested value 1) and the peer_not_allowed_to_send (suggested value 2) modes. The values 0 and 255 should be reserved. This registry uses the Specification Required policy as described in [RFC5226]. The reference should be RFCXXXX.

7. Security Considerations

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

8. Acknowledgments

The authors wish to thank Pasi Eronen, Adam Langley, Tom Petch, Eric Rescorla, Peter Saint-Andre, and Juho Vaehae-Herttua for their invaluable comments.

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.

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

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

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

[RFC6066] Eastlake, D., "Transport Layer Security (TLS) Extensions: Extension Definitions", RFC 6066, January 2011.

9.2. Informative References

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

[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.

[RFC5238] Phelan, T., "Datagram Transport Layer Security (DTLS) over the Datagram Congestion Control Protocol (DCCP)", RFC 5238, May 2008.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, January 2011.

Authors' Addresses

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

Email: seggelmann@fh-muenster.de

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

Email: tuexen@fh-muenster.de

Michael Williams

Email: michael.glenn.williams@gmail.com