RFC 6520 : Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 8 février 2012. Dernière mise à jour le 8 avril 2014

Date de publication du RFC : Février 2012

https://www.bortzmeyer.org/6520.html

Le protocole de cryptographie TLS ne disposait pas en standard de mécanisme de « battement de cœur », permettant de tester qu'une connexion est toujours vivante. C'est désormais chose faite. Grâce à la nouvelle extension de TLS, normalisée dans ce RFC, deux pairs TLS peuvent s'assurer, par un mécanisme standard, que l'autre pair est bien là, et répond.

Savoir si l'autre partie de la connexion est toujours là est souvent utile en matière de réseau. Les sessions qui utilisaient TLS (RFC 5246 [1]) ou DTLS (RFC 6347) ne disposaient pas d'un moyen générique de tester si le cœur du pair battait toujours. Les applications devaient donc envoyer des données inutiles et attendre une réponse. Une autre solution était de faire une renégociation (RFC 5246, section 7.4.1) mais cette opération est coûteuse en calculs cryptographiques. C'était encore pire avec DTLS, qui est parfois utilisé dans des applications unidirectionnelles, où aucune réponse applicative n'est prévue.

Désormais, les mises en œuvre de TLS qui gèrent la nouvelle extension, `Heartbeat`, pourront envoyer des messages `HeartbeatRequest`, auxquels le pair devra répondre avec un `HeartbeatResponse`. Cette extension est spécifiée au moment de la négociation (section 2), dans les messages `Hello` (RFC 5246, section 7.4.1). Dans la syntaxe vaguement analogue à ASN.1 de TLS, la nouvelle extension est décrite par :

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

struct {
   HeartbeatMode mode;
} HeartbeatExtension;
```

1. Pour voir le RFC de numéro NNN, `https://www.ietf.org/rfc/rfcNNN.txt`, par exemple `https://www.ietf.org/rfc/rfc5246.txt`

Le mode `peer_not_allowed_to_send` est là pour indiquer qu'une machine veut utiliser l'extension pour tester son pair mais qu'elle-même ne répondra pas aux requêtes `HeartbeatRequest`. La nouvelle extension est enregistrée à l'IANA <https://www.iana.org/assignments/tls-extensiontype-va tls-extensiontype-values.xml> (numéro 15).

Une fois que les deux parties se sont mises d'accord, quel est le protocole pour l'utilisation de cette extension (section 3) ? Le test des battements de cœur fonctionne au-dessus du "*Record Protocol*" de TLS, auquel il ajoute deux messages :

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

Un message `HeartbeatRequest` peut arriver à tout moment pendant une session, le pair doit alors répondre par un `HeartbeatResponse`. D'autres types que Requête et Réponse seront peut-être un jour créés, et enregistrés dans le nouveau registre IANA <https://www.iana.org/assignments/tls-parameters/tls-parameters.xml#heartbeat-message-types>.

Lorsque la session protégée par TLS fonctionne au dessus d'un protocole fiable, comme TCP, c'est tout. La couche de transport se chargera des éventuelles retransmissions. Si la session TLS utilise un protocole non-fiable comme UDP, l'émetteur du `HeartbeatRequest` doit se préparer, en l'absence de réponse, à réémettre le message (RFC 6347, section 4.2.4).

Et le contenu des messages (section 4) ? Une charge utile est autorisée :

```
struct {
     HeartbeatMessageType type;
     uint16 payload_length;
     opaque payload[HeartbeatMessage.payload_length];
     opaque padding[padding_length];
  } HeartbeatMessage;
```

Le contenu du champ `payload` est quelconque. La réponse doit inclure le même contenu (elle doit être un écho de la requête).

La section 5 du RFC décrit ensuite deux cas d'utilisation pratique de cette extension. Le premier est celui de la découverte de la MTU du chemin, pour laquelle DTLS n'avait pas de mécanisme (RFC 6347, section 4.1.1.1). La méthode, inspirée du RFC 4821, est simplement d'utiliser des `HeartbeatRequest` de plus en plus gros, jusqu'à ce qu'on n'obtienne plus de réponse, auquel cas on sait qu'on a trouvé la MTU maximale.

L'autre usage de cette extension est le test de bonne santé du pair, comme expliqué au début de cet article. Le RFC recommande de ne pas en abuser, et de n'émettre les `HeartbeatRequest` qu'à des intervalles de plusieurs fois le RTT.

Il existe déjà au moins une mise en œuvre de ce mécanisme de battement de cœur. OpenSSL l'aura à partir de la 1.0.1 (le code est déjà dans les versions de développement, l'extension - option `heartbeats` - est apparemment compilée par défaut). Pour GnuTLS, ce n'est encore qu'un projet. Comme souvent en informatique, quand on ajoute du code, on ajoute des bogues. C'est ainsi que la mise en œuvre de cette extension dans OpenSSL a été à l'origine de la faille CVE-2014-0160, dite « "*HeartBleed*" <http://heartbleed.com/> ».