RFC 6733 : Diameter Base Protocol

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 26 octobre 2012

Date de publication du RFC : Octobre 2012

https://www.bortzmeyer.org/6733.html

Pendant longtemps, le seul protocole standard d'authentification entre un NAS et sa base d'utilisateurs était Radius. Diameter, normalisé dans notre RFC (qui fait suite au RFC 3588 [1]), visait à le remplacer, et est bien plus riche en fonctions.

Traditionnellement, lorsqu'un client PPP ou d'un protocole similaire se connecte à son FAI, sa session est terminée sur un NAS, équipement qui ne possède en général pas de base des utilisateurs autorisés. Il y a plusieurs raisons à cela, comme le fait que le NAS n'est en général pas un ordinateur généraliste, il a peu de moyens, notamment en place disque ou bien comme le fait qu'un FAI a typiquement plusieurs NAS (au moins un par POP), et qu'il souhaite centraliser l'authentification. (Dans le monde 802.11, le NAS est typiquement nommé AP - "*Access Point*".)

Le mécanisme habituel est donc que le NAS soit **client** d'un protocole d'authentification et que le FAI gère un **serveur** de ce protocole, serveur qui connaitra les utilisateurs autorisés. Le même protocole assure en général également des fonctions de comptabilisation des accès, d'où le sigle AAA ("*Authentication, Authorization and Accounting*"). Le premier protocole standard en ce sens est Radius (RFC 2865), que Diameter vise à remplacer. (Un protocole privé, Tacacs, qui sera documenté dans le RFC 1492, avait également été utilisé.) Diameter avait été normalisé à l'origine dans le RFC 3588. Après une longue gestation (plusieurs années, alors qu'aucun changement de fond n'a été fait, et pas moins de trois « "*Last Calls*" » devant l'IETF), il est désormais remplacé par ce nouveau RFC 6733. Le protocole reste le même et les logiciels conformes à l'ancien et au nouveau RFC devraient interopérer.

Que reproche Diameter à Radius ? La section 1 de notre RFC détaille les faiblesses de Radius perçues par les auteurs de Diameter. Une analyse plus détaillée figure dans le RFC 3169 :

— Utilisation d'UDP, protocole de transport n'assurant pas de fiabilité,
— Pas de mécanisme de redondance standard,

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc3588.txt

— Limites de taille pour certains paramètres (par exemple la longueur d'un attribut n'est codée que sur un octet, ce qui limite les attributs à 256 caractères).

Diameter est donc plus riche et plus complexe que Radius. Diameter reprend aussi plusieurs idées de Radius comme le codage en doublets Attribut / Valeur (AVP pour "*Attribute-Value Pairs*") ou comme l'utilisation des NAI du RFC 7542 pour coder l'identité vérifiée.

Le RFC est bien plus long et compliqué que son équivalent Radius (154 pages !). La section 1 introduit le protocole, et ses usages, et le compare aux exigences listées dans le RFC 2989.

La section 2 précise le protocole, par exemple le mécanisme de transport fait l'objet de la sous-section 2.1, qui impose au minimum TCP et SCTP (les détails sont dans le RFC 3539). Diameter écoute par défaut sur le port 3868. La 2.4 introduit le nouveau concept d'"*Application Identifier*", valeur enregistrées `<https://www.iana.org/assignments/aaa-parameters>` auprès de l'IANA qui identifie un usage particulier de Diameter (plus riche que Radius, Diameter permet davantage d'usages). Par exemple, l'usage en tant que service d'authentification pour un NAS ("*Application Identifier*" n° 1) n'est pas dans le cœur du protocole, il est délégué au RFC 4005. D'autres RFC décrivent des applications de Diameter comme le RFC 4740 qui parle de son usage pour SIP. 2.5 parle de la différence entre connexions et sessions, une connexion étant un concept de la couche 4 alors qu'une session Diameter peut impliquer plusieurs connexions. En 2.8, on trouve le bestiaire des agents Diameter, les clients et les serveurs bien sûr mais également les relais ou bien les traducteurs, pour parler à des agents d'autres protocoles, notamment Radius.

La section 3 décrit l'en-tête des paquets Diameter et les codes de commande qu'ils contiennent comme `Capabilities-Exchange-Request`. Seuls les codes communs à toutes les applications sont définis dans la section 3.1, des codes aussi importants que `AA-Request` (demande d'authentification, l'équivalent du `Access-Request` de Radius) sont délégués à d'autres RFC (comme le RFC 4005).

En section 4, nous arrivons aux AVP eux-même, les doublets attributs-valeur qui portent l'information. On notera en 4.1 un concept désormais fréquent dans les protocoles réseaux, l'option "*Mandatory*" qui indique pour un AVP, s'il doit absolument être connu de l'implémentation (le dialogue Diameter avorte autrement) ou bien s'il peut tranquillement être ignoré, par exemple par une vieille implémentation qui ne connait pas ce nouvel AVP. La sous-section 4.3 décrit, entre autre, les URI Diameter, de plan `aaa:`, comme `aaa://host.example.com:666:transport=stcp;protocol=diameter`.

Les sections 5 à 8 spécifient les nombreuses machines à état qui doivent être mises en œuvre par les agents Diameter. Au contraire de Radius qui, comme le DNS, est purement requête/réponse, Diameter nécessite un vrai automate dans chaque agent.

La sous-section 5.2 explique comment un agent Diameter peut en trouver un autre. Avec Radius, la configuration (par exemple du serveur d'authentification) devait être manuelle. Avec Diameter, on peut utiliser SLP (RFC 2165) ou NAPTR combiné avec SRV (RFC 2782).

La section 13, sur la sécurité, est très stricte. Un agent Diameter doit pouvoir faire du TLS (mais, contrairement au RFC 3588, IPsec n'est plus qu'une possibilité secondaire).

Une bonne introduction à Diameter a été publié en `<http://www.interlinknetworks.com/whitepapers/Introduction_to_Diameter.pdf>`. Il n'existe que trois implémentations de serveur Diameter en logiciel libre, Free Diameter `<http://www.freediameter.net/>`, Open Diameter `<http://www.opendiameter.org/>` et OpenBlox Diameter `<http://www.traffixsystems.com/solutions/traffix-openblox-diameter-stack/>`, et, à part la première, elles semblent peu maintenues. (Et je ne suis pas sûr qu'OpenBlox soit vraiment libre.)

Le bilan opérationnel de Diameter est mince : beaucoup plus complexe que Radius, il ne l'a pas réellement remplacé dans son secteur, l'authentification via un NAS. Diameter semble bien être une nouvelle victime de l'effet « Deuxième système ».

Quels sont les changements par rapport au premier RFC Diameter, le RFC 3588 ? La section 1.1.3 en donne un résumé (la liste exacte est trop longue pour être incluse dans ce RFC déjà très long). Si les changements sont nombreux, ils ne sont pas suffisamment importants pour qu'on puisse parler d'un nouveau protocole. C'est le même Diameter et les nouveaux logiciels pourront parler avec les anciens. Les changements sont des clarifications, des abandons de certaines fonctions, et la correction des nombreuses erreurs du RFC précédent <http://www.rfc-editor.org/errata_search.php?rfc=3588&rec_status=15&presentation=table> (à mon avis, il y a une leçon à en tirer : les spécifications compliquées sont plus difficiles à faire correctement). Certaines fonctions étaient mentionnées dans le RFC 3588 mais même pas décrites complètement. Les principaux changements :

— Plus de mécanisme pour négocier en cours de session une sécurisation de la session par le chiffrement et introduction d'un nouveau port, le 5868, pour des connexions chiffrées avec TLS dès le début. (En fait, l'ancien mécanisme reste mais uniquement pour assurer la compatibilité.)
— La demande irréaliste du RFC 3588 d'avoir toujours IPsec a été retirée.
— Simplification de plusieurs fonctions comme la découverte du pair (pour laquelle il existait un zillion de méthodes dans le RFC 3588).
— Et bien d'autres clarifications, corrections de bogue dans les machines à état, etc.