RFC 9063 : Host Identity Protocol Architecture

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 15 juillet 2021

Date de publication du RFC : Juillet 2021

https://www.bortzmeyer.org/9063.html

Ce RFC propose d'aborder l'architecture de l'Internet en utilisant un nouveau type d'identificateur, le "*Host Identifier*" (HI), pour beaucoup d'usages qui sont actuellement ceux des adresses IP. Il remplace le RFC 4423 [1], qui était la description originale du protocole HIP, mais il n'y a pas de changements fondamentaux. HIP était un projet très ambitieux mais, malgré ses qualités, la disponibilité de plusieurs mises en œuvre, et des années d'expérimentation, il n'a pas percé.

Une adresse IP sert actuellement à deux choses : désigner une machine (l'adresse IP sert par exemple à distinguer plusieurs connexions en cours) et indiquer comment la joindre (routabilité). Dans le premier rôle, il est souhaitable que l'adresse soit relativement permanente, y compris en cas de changement de FAI ou de mobilité (actuellement, si une machine se déplace et change d'adresse IP, les connexions TCP en cours sont cassées). Dans le second cas, on souhaite au contraire une adresse qui soit le plus « physique » possible, le plus dépendante de la topologie. Ces deux demandes sont contradictoires.

HIP résout le problème en séparant les deux fonctions. Avec HIP, l'adresse IP ne serait plus qu'un identifiant « technique », ne servant qu'à joindre la machine, largement invisible à l'utilisateur et aux applications (un peu comme une adresse MAC aujourd'hui). Chaque machine aurait un HI ("*Host Identifier*") unique. Contrairement aux adresses IP, il n'y a qu'un HI par machine "*multi-homée*" mais on peut avoir plusieurs HI pour une machine si cela correspond à des usages différents, par exemple une identité publique, et une « anonyme ».

Pour pouvoir être vérifié, le nouvel identificateur, le HI sera (dans la plupart des cas) une clé publique cryptographique, qui sera peut-être allouée hiérarchiquement par PKI ou plutôt de manière répartie par tirage au sort (comme le sont les clés SSH ou PGP aujourd'hui, ce qui serait préférable, question vie privée). Ces identificateurs fondés sur la cryptographie permettent l'authentification réciproque des

1. Pour voir le RFC de numéro NNN, https://www.ietf.org/rfc/rfcNNN.txt, par exemple https://www.ietf.org/rfc/rfc4423.txt

machines (contrairement à IP, où il est trivial de mentir sur son adresse), et d'utiliser ensuite IPsec (RFC 7402) pour chiffrer la communication (HIP n'impose pas IPsec, plusieurs encapsulations des données sont possibles, et négociées dynamiquement, mais, en pratique, la plupart des usages prévus reposent sur IPsec).

L'authentification permet d'être sûr du HI de la machine avec qui on parle et, si le HI était connu préalablement à partir d'une source de confiance, d'être sûr qu'on parle bien à l'interlocuteur souhaité. (Si on ne connait pas le HI à l'avance, on dit que HIP est en mode « opportuniste ».)

Cette séparation de l'identificateur et du localisateur <https://www.bortzmeyer.org/separation-identi
html> est un sujet de recherche commun et d'autres propositions que HIP existent, comme LISP (RFC 6830) ou ILNP (RFC 6740). Dans tous les cas, les couches supérieures (comme TCP) ne verront que l'identificateur, permettant au localisateur de changer sans casser les sessions de transport en cours. (Un mécanisme ressemblant est le "*Back to My Mac*" du RFC 6281.) L'annexe A.1 de notre RFC rappelle les avantages de cette approche. Et l'annexe A.2, lecture très recommandée, note également ses défauts, l'indirection supplémentaire ajoutée n'est pas gratuite, et entraine des nouveaux problèmes. Notamment, il faut créer un système de correspondance - "*mapping*" - entre les deux, système qui complexifie le projet. Il y a aussi la latence <https://www.bortzmeyer.org/latence.html> supplémentaire due au protocole d'échange initial, qui est plus riche. Comparez cette honnêteté avec les propositions plus ou moins pipeau <https://www.bortzmeyer.org/table-rase-et-john-day.html> de « refaire l'Internet en partant de zéro », qui ne listent jamais les limites et les problèmes de leurs solutions miracle.

Ce HI ("*Host Identifier*") pourra être stocké dans des annuaires publics, comme le DNS (RFC 8005), ou une DHT (RFC 6537), pour permettre le rendez-vous (RFC 8004) entre les machines.

Notez que ce n'est pas directement le "*Host Identifier*", qui peut être très long, qui sera utilisé dans les paquets IP, mais un condensat, le HIT ("*Host Identity Tag*").

HIP intègre les leçons de l'expérience avec IP, notamment de l'importance d'authentifier la machine avec qui on parle. C'est ce qui est fait dans l'échange initial qui permet à un initiateur et un répondeur de se mettre à communiquer. Notamment, il y a obligation de résoudre un puzzle cryptographique, pour rendre plus difficile certaines attaques par déni de service. Voir à ce sujet « "*DOS-Resistant Authentication with Client Puzzles*" <http://www.hashcash.org/papers/dos-client-puzzles.
pdf> » de Tuomas Aura, Pekka Nikander et Jussipekka Leiwo, « "*Deamplification of DoS Attacks via Puzzles*" <https://pdfs.semanticscholar.org/fcd2/b361156ca06acab73724584653cea3d9ab02.
pdf> » de Jacob Beal et Tim Shepard ou encore « "*Examining the DOS Resistance of HIP*" <https:
//eprints.qut.edu.au/10145/> » de Tritilanunt, Suratose, Boyd, Colin A., Foo, Ernest, et Nieto, Juan Gonzalez.

La sécurité est un aspect important de HIP. Les points à garder en tête sont :

— Protection contre certaines attaques par déni de service via le puzzle cryptographique à résoudre.
— Protection contre les attaques de l'homme du milieu si le HI a été obtenu par un mécanisme sûr. Cela ne marche évidemment pas en mode opportuniste, où l'initiateur ne connait pas le HI de son correspondant, et le découvre une fois la connexion faite.
— Le mode opportuniste peut être renforcé, question sécurité, par le TOFU (RFC 7435).
— Tout mécanisme de séparation de l'identificateur et du localisateur ouvre de nouveaux problèmes : que se passe-t-il si le correspondant ment sur son localisateur ? Est-ce que cela permet des attaques par réflexion <https://www.bortzmeyer.org/attaques-reflexion.html> ? C'est pour éviter cela que les systèmes à séparation de l'identificateur et du localisateur prévoient, comme HIP, un test de l'existence d'une voie de retour <https://www.bortzmeyer.org/returnability.
html> (Cf. RFC 4225).

— Enfin, on peut évidemment mettre des ACL sur des HI mais leur structure « plate » fait qu'il n'y a pas d'agrégation possible de ces ACL (il faut une ACL par machine avec qui on correspond).

Au sujet du TOFU, le RFC cite « " *Leap-of-faith security is enough for IP mobility*" <https://dl.acm.org/citation.cfm?id=1700740> » de Miika Kari Tapio Komu et Janne Lindqvist, « "*Security Analysis of Leap-of-Faith Protocols*" <https://link.springer.com/chapter/10.1007/978-3-642-31909-9_19> » de Viet Pham et Tuomas Aura et « "*Enterprise Network Packet Filtering for Mobile Cryptographic Identities*" <https://www.usenix.org/legacy/event/usenix07/posters/lindqvist.pdf> » de Janne Lindqvist, Essi Vehmersalo, Miika Komu et Jukka Manner.

Notre RFC ne décrit qu'une architecture générale, il est complété par les RFC 7401, qui décrit le protocole, RFC 7402, RFC 8003, RFC 8004, RFC 8005, RFC 8046 et RFC 5207. Si des implémentations expérimentales existent déjà et que des serveurs publics utilisent HIP, aucun déploiement significatif n'a eu lieu (cf. l'article « "*Adoption barriers of network layer protocols : The case of host identity protocol*" de T. Leva, M. Komu, A. Keranen et S. Luukkainen). Comme le disait un des relecteurs du RFC, « "*There's a lot of valuable protocol design and deployment experience packed into this architecture and the associated protocol RFCs. At the same time, actual adoption and deployment of HIP so far appears to have been scarce. I don't find this surprising. The existing Internet network/transport/application protocol stack has already become sufficiently complicated that considerable expertise is required to manage it in all but the simplest of cases. Teams of skilled engineers routinely spend hours or days troubleshooting operational problems that crop up within or between the existing layers, and the collection of extensions, workarounds, identifiers, knobs, and failure cases continues to grow. Adding a major new layer–and a fairly complicated one at that–right in the middle of the existing stack seems likely to explode the already heavily-strained operational complexity budget of production deployments.*" ». L'annexe A.3 décrit les questions pratiques liées au déploiement. Elle rappelle le compte-rendu d'expérience chez Boeing de Richard Paine dans son livre « "*Beyond HIP : The End to Hacking As We Know It*" <https://dl.acm.org/citation.cfm?id=1823457> ». Elle tord le cou à certaines légendes répandues (que HIP ne fonctionne pas à travers les routeurs NAT, ou bien qu'il faut le mettre en œuvre uniquement dans le noyau.)

Ah, question implémentations (RFC 6538), on a au moins HIP for Linux <http://mkomu.kapsi.fi/hipl/> et OpenHIP <http://openhip.sourceforge.net/> qui ont été adaptés aux dernières versions de HIP, et des protocoles associés.

Les changements depuis le RFC 4423 sont résumés en section 14. Il n'y en a pas beaucoup, à part l'intégration de l'expérience, acquise dans les treize dernières années (et résumée dans le RFC 6538) et des améliorations du texte. La nouvelle annexe A rassemble plein d'informations concrètes, notamment que les questions pratiques de déploiement de HIP, et sa lecture est très recommandée à tous ceux et toutes celles qui s'intéressent à la conception de protocoles. La question de l'agilité cryptographique (RFC 7696) a également été détaillée.