Groupe de travail Réseau
Request for Comments: 5535
Catégorie : Sur la voie de la normalisation

M. Bagnulo, UC3M
juin 2009
Traduction Claude Brière de L'Isle

Adresses fondées sur le hachage (HBA)

Statut de ce mémoire

Le présent document spécifie un protocole sur la voie de la normalisation de l'Internet pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Normes officielles des protocoles de l'Internet" (STD 1) pour connaître l'état de la normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Résumé

Le présent mémoire décrit un mécanisme pour fournir un lien sûr entre les diverses adresses avec les différents préfixes disponibles à un hôte au sein d'un site multi-rattachements. Ce mécanisme emploie soit des adresses générées cryptographiquement (CGA, *Cryptographically Generated Address*) soit une nouvelle variante du même thème qui utilise le même format dans les adresses. L'idée principale dans la nouvelle variante est que les informations sur les multiples préfixes sont incluses dans les adresses elles mêmes. Cela est réalisé en générant les identifiants d'interface des adresses d'un hôte comme des hachages des préfixes disponibles et d'un nombre aléatoire. Ensuite, les diverses adresses sont générées en ajoutant devant, les différents préfixes aux identifiants d'interface générés. Le résultat est un ensemble d'adresses, appelées adresses fondées sur le hachage (HBA, *Hash-Based Address*) qui sont par nature liées les unes aux autres.

Table des matières

1. Introduction

Afin de préserver l'adaptabilité du système d'acheminement inter-domaines, les sites IPv6 obtiennent des adresses de la part de leur fournisseurs d'accès Internet (FAI). Une telle stratégie d'adressage réduit significativement la quantité de chemins dans les tableaux d'acheminement mondiaux, car chaque FAI annonce seulement les chemins de ses propres blocs d'adresses, plutôt que d'annoncer un chemin par site consommateur. Cependant, ce schéma d'adressage implique que les sites multi rattachements vont obtenir plusieurs préfixes, un par FAI. De plus, comme chaque FAI annonce seulement son propre bloc d'adresses, un site multi rattachements va être accessible à travers un FAI donné si le préfixe du FAI est contenu dans l'adresse de destination des paquets. Cela signifie que, si une communication établie doit être acheminée à travers différents FAI durant sa durée de vie, des adresses avec des préfixes différents auront été utilisées. Changer l'adresse utilisée pour porter les paquets d'une communication établie expose la communication à de nombreuses attaques, comme décrit dans la [RFC4218], de sorte que des mécanismes de sécurité sont nécessaires pour fournir la protection requise aux parties impliquées. Le présent mémoire décrit un outil qui peut être utilisé pour fournir la protection contre certaines des attaques potentielles, en particulier contre des attaques futures/préméditées (autrement dit des attaques en temps glissant [RFC4225]).

Le présent mémoire décrit un mécanisme pour fournir un lien sûr entre les multiples adresses avec différents préfixes disponibles à un hôte au sein d'un site multi rattachements.

On devrait noter que, à la différence du cas de la mobilité où les adresses qui vont être utilisées par le nœud mobile ne sont pas connues a priori, les multiples adresses disponibles à un hôte dans le site multi rattachements sont pré-définies et connues à l'avance dans la plupart des cas. Le mécanisme proposé dans ce mémoire emploie soit des adresses générées cryptographiquement (CGA, *Cryptographically Generated Address*) [RFC3972] soit une nouvelle variante du même thème qui utilise le même format dans les adresses. La nouvelle variante, l'adresse fondée sur le hachage (HBA, *Hash-Based Address*) tire parti de la stabilité de l'ensemble d'adresses. Dans les deux cas, un lien sûr entre les adresses d'un nœud dans un site multi rattachements peut être fourni. Les CGA emploient la cryptographie de clé publique et peuvent traiter les changements d'ensemble d'adresses. Les HBA emploient seulement la cryptographie de clés symétriques, et ont de plus faibles exigences de calcul.

Pour les besoins du protocole Shim6, les autres caractéristiques des CGA et des HBA sont similaires. Toutes deux peuvent être générées par l'hôte lui-même sans aucun support d'une infrastructure externe. Toutes deux emploient le même format d'adresses et le même format de données pour générer les adresses. Il n'est pas exigé que tous les identifiants d'interface des adresses d'un nœud soient égales, préservant un certain degré de confidentialité à travers les changements des adresses utilisées durant les communications.

L'idée principale des HBA est que les informations sur les multiples préfixes sont incluses dans les adresses elles-mêmes. Ceci est réalisé en générant les identifiants d'interface des adresses d'un hôte comme des hachages des préfixes disponibles et d'un nombre aléatoire. Ensuite les multiples adresses sont obtenues en ajoutant les différents préfixes devant les identifiants d'interface générés. Le résultat est un ensemble d'adresses qui sont liées naturellement. Un mécanisme de faible coût est disponible pour déterminer si deux adresses appartiennent au même ensemble, car étant donné l'ensemble de préfixes et les paramètres supplémentaires utilisés pour générer la HBA, une seule opération de hachage est suffisante pour vérifier si une HBA appartient à un ensemble de HBA donné.

2. Terminologie

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119],.

3. Vue d'ensemble

3.1 Modèle de menace

L'analyse des menaces pour le problème du multi rattachements est décrit dans la [RFC4218]. Cette analyse identifie les attaques de base fondées sur la redirection des paquets par un attaquant malveillant vers des adresses qui n'appartiennent pas au nœud multi rattaché. Il y a essentiellement deux types d'attaques de redirection : la capture de communication et les attaques d'inondation. Les attaques de capture de communication sont celles où un attaquant vole des communications en cours et/ou futures à une victime. Les attaques d'inondation sont celles où on redirige le trafic généré par une source légitime vers un tiers, pour l'inonder. La solution HBA donne une protection complète contre les attaques de capture de communication. Le protocole Shim6 [RFC5533] protège contre les attaques d'inondation. Les menaces résiduelles sont décrites dans la section "Considérations sur la sécurité".

3.2 Généralités

L'objectif de base du mécanisme de HBA est de lier en toute sécurité plusieurs adresses IPv6 qui appartiennent au même hôte multi rattaché. Cela permet de réacheminer le trafic sans craindre que la communication soit redirigée sur un attaquant. La technique qui est utilisée est d'inclure un hachage des préfixes permis dans les bits de moindre poids de l'adresse IPv6.

Donc, en éludant certains détails, disons que les préfixes disponibles sont A, B, C, et D, que l'hôte va générer une liste de préfixes P consistant en (A,B,C,D) et un nombre aléatoire appelé le modificateur M. Ensuite il va générer les nouvelles adresses :

A || H(M || A || P)
B || H(M || B || P)
C || H(M || C || P)
D || H(M || D || P)

Donc, avec une adresse valide sur le groupe, la liste de préfixes P, et le modificateur aléatoire M, il est possible de déterminer si une autre adresse fait partie du groupe en calculant le hachage et en le confrontant aux bits de moindre poids.

3.3 Motivations du concept de HBA

La conception de la technique de HBA a été conduite par les considérations suivantes :

Tout d'abord, le but de HBA est de fournir un lien sûr entre l'adresse IPv6 utilisée comme un identifiant par les protocoles de couche supérieure et les localisateurs de remplacement disponibles dans le nœud multi rattaché afin d'empêcher les attaques en redirection.

Ensuite, afin de réaliser une telle protection, l'approche choisie a été d'inclure les informations de sécurité dans l'identifiant lui-même, au lieu de s'appuyer sur des tiers de confiance pour sécuriser le lien, comme ceux fondés sur des répertoires ou une infrastructure de clés publiques. Cette décision a été conduite par des considérations de déploiement, c'est-à-dire, le coût de déploiement de l'infrastructure de tiers de confiance.

Troisièmement, les considérations de prise en charge d'application décrites dans [16] ont résulté en le choix d'adresses IPv6 acheminables à utiliser comme identifiants. Donc, les informations de sécurité sont insérées dans la partie identifiant d'interface de l'adresse IPv6.

Quatrièmement, des considérations de performances comme décrites dans [17] ont motivé l'usage d'une approche fondée sur le hachage plutôt que d'une approche fondée sur de pures adresses générées cryptographiquement (CGA, *Cryptographic Generated Address*) afin d'éviter d'imposer les performances des opérations de clé publique pour chaque communication dans les environnements multi rattachements. L'approche HBA présentée dans ce document offre une solution de remplacement moins coûteuse qui est intéressante pour de nombreux cas d'usage courants. Noter que l'approche HBA et l'approche CGA ne s'excluent pas mutuellement et qu'il est possible de générer des adresses qui soient à la fois des

CGA valides et des HBA qui fournissent les avantages des deux approches si nécessaire.

4. Considérations sur la compatibilité des adresses générées cryptographiquement (CGA)

Comme décrit à la section précédente, la technique HBA utilise la partie identifiant d'interface de l'adresse IPv6 pour coder les informations sur les multiples préfixes disponibles à un hôte multi rattaché. Cependant, l'identifiant d'interface est aussi utilisé pour porter des informations cryptographiques quand des adresses générées cryptographiquement (CGA) [RFC3972] sont utilisées. Donc, des conflits d'usage des bits d'identifiant d'interface peuvent résulter si ce n'est pas pris en compte durant la conception de HBA. Il y a au moins deux raisons valides de fournir la compatibilité CGA-HBA :

D'abord, la spécification actuelle de découverte de voisin sûre (SeND) [RFC3971] utilise les CGA définies dans la [RFC3972] pour prouver la possession de l'adresse. Si les HBA ne sont pas compatibles avec les CGA, alors les nœuds qui utilisent les HBA pour le multi rattachement ne seraient pas capables de faire la découverte de voisin sûre en utilisant les mêmes adresses (au moins les parties de SeND qui exigent des CGA). Cela impliquerait que les nœuds devraient choisir entre sécurité (provenant de SeND) et tolérance aux fautes (provenant de la prise en charge de multi rattachements IPv6 fournie par le protocole Shim6 [RFC5533]). De plus pour SeND, il y a d'autres protocoles qui sont considérés tirer parti des avantages offerts par le schéma de CGA, comme les protocoles de prise en charge de la mobilité [RFC4866]. Ces protocoles ne pourraient pas être utilisés avec des HBA si les HBA n'étaient pas compatibles avec les CGA.

Ensuite, les CGA fournissent des caractéristiques supplémentaires qui ne peuvent pas être réalisées en utilisant seulement des HBA. En particulier, parce que de sa propre nature, la technique de HBA prend seulement en charge un ensemble de préfixes prédéterminé qui est connu au moment de la génération de l'ensemble de HBA. Aucun ajout de nouveaux préfixes à cet ensemble original n'est accepté après la génération de l'ensemble de HBA. Dans la plupart des cas pertinents pour le multi-rattachements de site, ceci n'est pas un problème parce que l'ensemble de préfixes disponibles à un ensemble multi rattaché n'est pas très dynamique. De nouveaux préfixes peuvent être ajoutés dans un site multi rattachements quand un nouveau FAI est disponible, mais le rythme de ces événements est rarement à la même échelle de temps que la durée de vie des communications établies. Il est alors suffisant pour de nombreuses situations que le nouveau préfixe ne soit pas disponible pour les communications établies et que seules les nouvelles communications en bénéficient. Cependant, dans le cas où une telle fonction est requise, il est possible d'utiliser des CGA pour la fournir. Cette approche exige clairement que les approches de HBA et CGA soient compatibles. Si c'est le cas, il serait alors possible de créer des adresses HBA/CGA qui prennent en charge simultanément la fonction de CGA et de HBA. Les entrées au processus de génération de HBA/CGA vont être un ensemble de préfixes et une clé publique. De cette façon, un nœud qui a établi une communication en utilisant une adresse de l'ensemble de CGA/HBA peut dire à son homologue d'utiliser la vérification de HBA quand une des adresses de son ensemble de HBA/CGA est utilisée comme localisateur dans la communication ou d'utiliser la vérification de CGA (fondée sur la clé publique/privée) quand une nouvelle adresse qui n'appartient pas à l'ensemble de HBA/CGA est utilisée comme localisateur dans la communication.

Donc, pour les raisons susmentionnées, il est un but de la conception de HBA de définir les HBA d'une façon telle qu'elles soient compatibles avec les CGA comme défini dans la [RFC3972] et leurs usages décrits dans la [RFC3971] (par conséquent, pour comprendre le reste de cette note, le lecteur devrait être familiarisé avec la spécification de CGA définie dans la [RFC3972]). Cela signifie qu'il doit être possible de générer des adresses qui soient à la fois des HBA et des CGA, c'est-à-dire, que l'identifiant d'interface contienne des informations cryptographiques de CGA et les informations d'ensemble de préfixes d'une HBA. La spécification de CGA considère déjà la possibilité d'inclure des informations supplémentaires dans le processus de génération de CGA par l'usage des champs Extension dans la structure de données de paramètre de CGA. Il est alors possible de définir une extension Multi-préfixes pour la CGA afin que les informations d'ensemble de préfixes soient incluses dans le processus de génération d'identifiant d'interface.

Bien qu'une approche compatible à la CGA soit adoptée, on devrait noter que HBA et CGA sont des concepts différents. En particulier, la CGA est liée de façon inhérente à une clé publique, alors qu'une HBA est liée de façon inhérente à un ensemble de préfixes. Cela signifie qu'une clé publique n'est pas exigée pour générer une adresse seulement HBA. À cause de cela, on définit trois différents types d'adresses :

- adresses seulement CGA : ce sont des adresses générées comme spécifié dans la [RFC3972] sans inclure l'extension Multi-préfixes. Elles sont liées à une clé publique et à un seul préfixe (contenu dans la structure de données de paramètre de CGA de base). Ces adresses peuvent être utilisées pour SeND [RFC3971] ; si elles sont utilisées pour le multi rattachements, leur application va devoir être fondée sur l'usage de la clé publique.

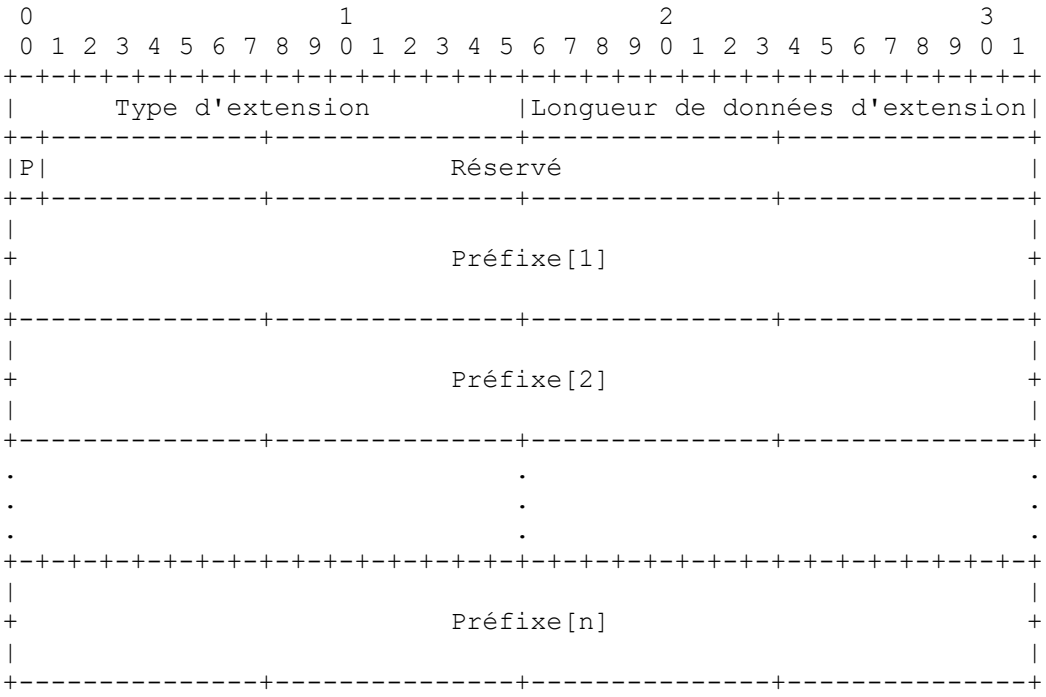
- adresses CGA/HBA : ces adresses sont des CGA qui incluent l'extension Multi-préfixes dans la structure de données de paramètre de CGA utilisée pour leur génération. Ces adresses sont liées à une clé publique et un ensemble de préfixes et

elles fournissent les deux fonctions de CGA et de HBA. Elles peuvent être utilisées pour SeND comme défini dans la [RFC3971] et pour tout usage défini pour HBA (comme un protocole Shim6).

- adresses seulement HBA : ces adresses sont liées à un ensemble de préfixes mais elles ne sont pas liées à une clé publique. Parce que les HBA sont compatibles avec les CGA, la structure de données de paramètre de CGA va être utilisée pour leur génération, mais un nom occasionnel aléatoire va être inclus dans le champ Clé publique au lieu d'une clé publique. Ces adresses peuvent être utilisées pour les protocoles de multi rattachements fondés sur la HBA, mais elles ne peuvent pas être utilisées pour SeND.

5. Extension multi-préfixes pour CGA

L'extension multi-préfixes a le format de TLV défini dans la [RFC4581] :



Type d'extension : identifiant de type de 16 bits de l'extension Multi-préfixes (voir la section "Considérations relatives à l'IANA").

Longueur de données d'extension : entier non signé de 16 bits. Longueur de l'extension en octets, non inclus les quatre premiers octets.

Fanion P : établi si une clé publique est incluse dans le champ Clé publique de la structure de données de paramètre de CGA, à zéro autrement.

Réservé : champ réservé de 31 bits. DOIT être initialisé à zéro, et ignoré à réception.

Préfixe[1...n] : vecteur de préfixes de 64 bits, numérotés de 1 à n.

6. Génération d'ensemble de HBA

Le processus de génération de HBA se fonde sur le processus de génération de CGA défini à la Section 4 de la [RFC3972]. Le but est d'exiger une quantité minimum de changements au processus de génération de CGA. On devrait noter que la procédure suivante n'est valide que pour les valeurs de Sec de 0, 1, et 2. Pour les autres valeurs de Sec, la [RFC4982] a défini un registre de SEC CGA qui va contenir les spécifications utilisées pour générer les CGA. Les procédures de génération définies dans ces spécifications doivent être utilisées pour les valeurs de Sec autres que 0, 1, ou 2.

Le processus de génération de CGA a trois entrées : un préfixe de sous réseau de 64 bits, une clé publique (codée en DER comme une structure ASN.1 de type SubjectPublicKeyInfo) et le paramètre de sécurité Sec.

La principale différence entre la génération de CGA et la génération de HBA est que alors qu'une CGA peut être générée de façon indépendante, toutes les HBA d'un ensemble donné de HBA doivent être générées en utilisant les mêmes paramètres, ce qui implique que la génération des adresses d'un ensemble de HBA va se faire de façon coordonnée. Dans le présent mémoire, on va décrire un mécanisme pour générer toutes les adresses d'un ensemble de HBA donné. Le processus de génération de chacune des adresses HBA d'un ensemble de HBA va être fondé sur le processus de génération de CGA défini dans la [RFC3972]. Plus précisément, le processus de génération d'ensemble de HBA va être défini comme une séquence légèrement modifiée de générations de CGA.

Les changements requis dans le processus de génération de CGA quand on génère une seule HBA sont les suivants : d'abord l'extension multi préfixes doit être incluse dans la structure de données de paramètre de CGA. Ensuite, dans le cas où l'adresse générée est seulement HBA, un nom occasionnel aléatoire va devoir être utilisé comme entrée à la place d'une clé publique valide. Pour les questions de rétro compatibilité avec les pures CGA, le nom occasionnel aléatoire DOIT être codé comme une clé publique comme défini dans la [RFC3972]. En particulier, le nom occasionnel aléatoire DOIT être formaté comme une structure ASN.1 codée en DER du type SubjectPublicKeyInfo, défini dans le profil de certificat Internet X.509 [RFC5280]. L'identifiant d'algorithme DOIT être rsaEncryption, qui est 1.2.840.113549.1.1.1, et le nom occasionnel aléatoire DOIT être formaté en utilisant le type RSAPublicKey comme spécifié au paragraphe 2.3.1 de la [RFC3279]. La longueur du nom occasionnel aléatoire est 384 bits.

Le processus résultant de génération d'ensemble de HBA est le suivant :

Les entrées au processus de génération de HBA sont :
- o un vecteur de n préfixes de 64 bits,
- o un paramètre Sec, et
- o dans le cas de la génération d'un ensemble d'adresses HBA/CGA, une clé publique est aussi fournie comme entrée (non exigée quand on génère des adresses seulement HBA).

Le résultat du processus de génération de HBA est :
- o un ensemble de HBA
- o leurs structures respectives de données de paramètre de CGA

Les étapes du processus de génération d'ensemble de HBA sont :

1. Génération de l'extension multi préfixes. Générer l'extension multi préfixes avec le format défini à la Section 5. Inclure le vecteur de n préfixes de 64 bits dans les champs Préfixe[1...n]. La valeur du champ Longueur d'extension est (n*8 + 4). Si une clé publique est fournie, alors le fanion P est réglé à un. Autrement, le fanion P est réglé à zéro.

2. Génération du modificateur. Générer un modificateur comme une valeur aléatoire ou pseudo aléatoire de 128 bits. Si une clé publique n'a pas été fournie en entrée, générer le modificateur étendu comme une valeur aléatoire ou pseudo aléatoire de 384 bits. Coder la valeur de modificateur étendu comme une clé RSA dans une structure ASN.1 codée en DER du type SubjectPublicKeyInfo défini dans le profil de certificat Internet X.509 [RFC5280].

3. Enchaîner de gauche à droite le modificateur, 9 octets de zéros, la clé publique codée ou le modificateur étendu codé (si aucune clé publique n'était fournie) et l'extension multi préfixes. Exécuter l'algorithme SHA-1 sur l'enchaînement. Prendre les 112 bits de gauche de la valeur du hachage SHA-1. Le résultat est Hash2.

4. Comparer les 16*Sec bits de gauche de Hash2 à zéro. Si ils sont tous zéro (ou si Sec=0) continuer à l'étape (5). Autrement, incrémenter le modificateur de un et revenir à l'étape (3).

5. Régler le compte de collisions de 8 bits à zéro.

6. Pour i=1 à n (nombre de préfixes) faire :

   6.1 Enchaîner de gauche à droite la valeur de modificateur final, Préfixe[i], le compte de collisions, la clé publique codée ou le modificateur étendu codé (si aucune clé publique n'était fournie) et l'extension multi préfixe. Exécuter l'algorithme SHA-1 sur l'enchaînement. Prendre les 64 bits de gauche de la valeur du hachage SHA-1. Le résultat est Hash1[i].

6.2 Former un identifiant d'interface à partir de Hash1[i] en écrivant la valeur de Sec dans les trois bits de gauche et en réglant les bits 6 et 7 (c'est-à-dire, les bits "u" et "g") à zéro.

6.3 Générer l'adresse HBA[i] en enchaînant Préfixe[i] et les 64 bits de l'identifiant d'interface pour former une adresse IPv6 de 128 bits avec le préfixe de sous réseau à gauche et l'identifiant d'interface à droite comme dans une adresse IPv6 standard [RFC4291].

6.4 Effectuer si nécessaire la détection d'adresse dupliquée. Si une collision d'adresse est détectée, incrémenter le compte de collisions de un et revenir à l'étape (6). Cependant, après trois collisions, arrêter et rapporter l'erreur.

6.5 Former la structure de données de paramètre de CGA qui correspond à HBA[i] en enchaînant de gauche à droite la valeur de modificateur final, Préfixe[i], la valeur du compte de collision finale, la clé publique codée ou le modificateur étendu codé, et l'extension multi préfixes.

Note : la plupart des étapes du processus sont tirées de la [RFC3972].

7. Vérification de HBA

La procédure suivante n'est valide que pour les valeurs de Sec de 0, 1, et 2. Pour les autres valeurs de Sec, la [RFC4982] a défini un registre de SEC CGA qui va contenir les spécifications utilisées pour vérifier les CGA. Les procédures de vérification définies dans de telles spécifications doivent être utilisées pour les valeurs de Sec autres que 0, 1, ou 2.

7.1 Vérification qu'une adresse HBA particulière correspond à une structure de données de paramètre CGA donnée

Les HBA sont construites comme une extension de CGA, donc une HBA correctement formatée et sa structure de données de paramètre de CGA correspondante vont terminer avec succès le processus de vérification décrit à la Section 5 de la [RFC3972]. Une telle vérification est utile quand le but est la vérification du lien entre la clé publique et la HBA.

7.2 Vérification qu'une adresse HBA particulière appartient à l'ensemble de HBA associé à une structure de données de paramètre CGA donnée

Pour les applications de multi rattachements, il est aussi pertinent que le receveur des informations de HBA vérifie si une certaine adresse HBA appartient à un certain ensemble de HBA. Un ensemble de HBA est identifié par une structure de données de paramètre de CGA qui contient une extension multi préfixes. Donc, le receveur doit vérifier si une certaine HBA appartient à l'ensemble de HBA défini par une structure de données de paramètre de CGA. On devrait noter que le receveur peut devoir vérifier si une HBA appartient à l'ensemble de HBA défini par la structure de données de paramètre de CGA d'une autre HBA de l'ensemble. Si c'est le cas, les HBA vont échouer au processus de vérification de CGA défini dans la [RFC3972], parce que le préfixe inclus dans le champ Préfixe de sous réseau de la structure de données de paramètre de CGA ne va pas correspondre au préfixe de la HBA à vérifier. Pour vérifier si une HBA appartient à un ensemble de HBA associé à une autre HBA, on vérifie que le préfixe de la HBA est inclus dans l'ensemble de préfixes défini dans l'extension multi préfixes, et si c'est le cas, on substitue alors au préfixe inclus dans le champ Préfixe de sous réseau le préfixe de la HBA, et on effectue ensuite le processus de vérification de CGA défini dans la [RFC3972].

Donc, le processus pour vérifier qu'une HBA appartient à un ensemble de HBA déterminé par une structure de données de paramètre de CGA est appelé la vérification de HBA et il est le suivant :

Les entrées au processus de vérification de HBA sont :
- une HBA
- une structure de données de paramètre de CGA

Les étapes du processus de vérification de HBA sont les suivantes :

1. Vérifier que le préfixe de HBA de 64 bits est inclus dans l'ensemble de préfixes de l'extension multi préfixes. Si il n'est pas inclus, la vérification échoue. Si il est inclus, remplacer le préfixe contenu dans le champ Préfixe de sous réseau de la structure de données de paramètre de CGA par le préfixe de 64 bits de la HBA.

2. Effectuer le processus de vérification décrit à la Section 5 de la [RFC3972] avec la HBA et la nouvelle structure de

données de paramètre de CGA (incluant l'extension multi préfixes) comme entrées. Les étapes du processus sont incluses ci-dessous, extraites de la [RFC3972] :

2.1 Vérifier que le compte de collisions dans la structure de données de paramètre de CGA est 0, 1, ou 2. La vérification de CGA échoue si le compte de collision est hors de la gamme valide.

2.2 Vérifier que le préfixe de sous réseau dans la structure de données de paramètre de CGA est égal au préfixe de sous réseau (c'est-à-dire, aux 64 bits de gauche) de l'adresse. La vérification de CGA échoue si les valeurs de préfixe diffèrent. Note : Cette étape réussit toujours à cause de l'action effectuée à l'étape 1.

2.3 Exécuter l'algorithme SHA-1 sur la structure de données de paramètre de CGA. Prendre les 64 bits de gauche de la valeur du hachage SHA-1. Le résultat est Hash1.

2.4 Comparer Hash1 avec l'identifiant d'interface (c'est-à-dire, les 64 bits de droite) de l'adresse. Les différences dans les trois bits de gauche et les bits 6 et 7 (c'est-à-dire, les bits "u" et "g") sont ignorées. Si les valeurs de 64 bits diffèrent (autres que dans les cinq bits ignorés) la vérification de CGA échoue.

2.5 Lire le paramètre de sécurité Sec dans les trois bits de gauche de l'identifiant d'interface de 64 bits de l'adresse. (Sec est un entier non signé de 3 bits.)

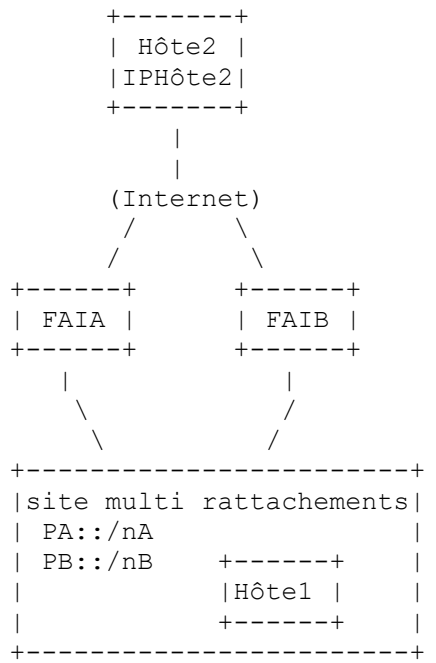
2.6 Enchaîner de gauche à droite le modificateur, 9 octets de zéros, la clé publique, et tous champs d'extension (dans ce cas, l'extension multi préfixes va au moins être incluse) qui suivent la clé publique dans la structure de données de paramètre de CGA. Exécuter l'algorithme SHA-1 sur l'enchaînement. Prendre les 112 bits de gauche de la valeur du hachage SHA-1. Le résultat est Hash2.

2.7 Comparer les 16*Sec bits de gauche de Hash2 à zéro. Si l'un d'eux n'est pas zéro, la vérification de CGA échoue. Autrement, la vérification réussit. (Si Sec=0, la vérification de CGA n'échoue jamais à cette étape.)

8. Exemple d'application HBA dans un scénario de multi-rattachements

Dans cette section, on va décrire une application possible de la technique de HBA au multi rattachements IPv6.

On va considérer le scénario suivant : un site multi rattachements obtient la connexité Internet à travers deux fournisseurs : FAIA et FAIB. Chaque fournisseur a délégué un préfixe au site multi rattachements (PrefA::/nA et PrefB::/nb, respectivement). Afin de bénéficier du multi rattachements, les hôtes au sein du site multi rattachements vont configurer plusieurs adresses IP, une par préfixe disponible. La configuration résultante est décrite dans la figure qui suit.



On suppose que Hôte1 et Hôte2 prennent tous deux en charge le protocole Shim6.

Hôte2 n'est pas situé dans un site multi rattachements, donc il n'a pas besoin de créer des HBA (il doit cependant être capable de les vérifier, afin de prendre en charge le protocole Shim6, comme on va le décrire ensuite).

Hôte1 est situé dans le site multi rattachements, donc il va générer ses adresses comme des HBA. Pour ce faire, il a besoin d'exécuter le processus de génération d'ensemble de HBA comme précisé à la Section 6 du présent mémoire. Les entrées du processus de génération d'ensemble de HBA vont être : un vecteur de préfixe contenant les deux préfixes disponibles dans sa liaison, c'est-à-dire, PA:LA::/64 et PB:LB::/64, une valeur de paramètre Sec, et facultativement une clé publique. Dans ce cas, on va supposer qu'une clé publique est fournie de sorte qu'on peut aussi illustrer comment un événement de renumérotage peut être pris en charge quand des adresses HBA/CGA sont utilisées (voir le paragraphe qui se réfère à la prise en charge dynamique de l'ensemble d'adresses). Donc, après l'exécution du processus de génération de l'ensemble de HBA, Hôte1 va avoir : un ensemble de HBA consistant en deux adresses, c'est-à-dire, PA:LA:iidA et PB:LB:iidB avec leurs structures respectives de données de paramètre de CGA, c'est-à-dire, CGA_PDS_A et CGA_PDS_B. Noter que iidA et iidB sont différents mais tous deux contiennent des informations sur l'ensemble de préfixes disponible dans le site multi rattachements.

On va ensuite considérer une communication entre Hôte1 et Hôte2. On suppose que les deux FAI du site multi rattachements fonctionnent correctement, de sorte que les adresses disponibles dans Hôte1 peuvent être utilisées pour la communication. On suppose ensuite que la communication est établie en utilisant PA:LA:iidA et IPHost2 pour Hôte1 et Hôte2, respectivement. Jusque là aucune prise en charge Shim6 particulière n'a été requise, et PA:LA:iidA est utilisé comme toute autre adresse IP mondiale.

Supposons qu'à un certain moment, un des hôtes impliqués dans la communication décide que la prise en charge du multi rattachements est nécessaire dans cette communication (cela signifie qu'un des hôtes impliqués dans la communication désire des capacités améliorés de tolérance aux fautes pour cette communication, afin que si une panne survient, la communication puisse être rattachée à un fournisseur de remplacement).

À ce moment, l'échange d'établissement de contexte de paire d'hôtes du protocole Shim6 va être effectué entre les deux hôtes (voir la [RFC5533]). Dans cet échange, Hôte1 va envoyer CGA_PDS_A à Hôte2.

Après la réception de CGA_PDS_A, Hôte2 va vérifier que la structure de données de paramètre de CGA reçue correspond à l'adresse utilisée dans la communication PA:LA:iidA. Cela signifie que Hôte2 va exécuter le processus de vérification de HBA décrit à la Section 7 de ce mémoire avec PA: LA:iidA et CGA_PDS_A comme entrées. Dans ce cas, la vérification va réussir parce que la structure de données de paramètre de CGA et les adresses utilisées dans la vérification correspondent.

Tant qu'il n'y a pas de panne affectant le chemin de communication à travers FAIA, les paquets vont continuer de s'écouler. Si une défaillance affecte le chemin à travers FAIA, Hôte1 va tenter de re-rattacher la communication à une adresse de remplacement, c'est-à-dire, PB:LB:iidB. Pour réaliser cela, après la détection de la panne, Hôte1 va informer Hôte2 de l'adresse de remplacement. Hôte2 va vérifier que la nouvelle adresse appartient à l'ensemble de HBA de l'adresse initiale. Pour faire cela, Hôte2 va exécuter le processus de vérification de HBA avec la structure de données de paramètre de CGA de l'adresse originale (c'est-à-dire, CGA_PDS_A) et la nouvelle adresse (c'est-à-dire, PB:LB:iidB) comme entrées. Le processus de vérification va réussir parce que PB:LB::/64 a été inclus dans l'extension multi préfixes durant le processus de génération d'ensemble de HBA. Des vérifications supplémentaires peuvent être nécessaires pour empêcher des attaques d'inondation (voir les commentaires sur la prévention des attaques d'inondation dans la section des considérations sur la sécurité du présent mémoire).

Une fois la nouvelle adresse vérifiée, elle peut être utilisée comme localisateur de remplacement pour rattacher la communication, tout en préservant l'adresse originale (PA:LA:iidA) comme identifiant pour les couches supérieures. Cela signifie que les paquets suivants vont être adressés de/vers cette nouvelle adresse. Noter qu'aucune vérification de HBA supplémentaire n'est exigée pour les paquets suivants, car la nouvelle adresse valide peut être mémorisée dans Hôte2.

Dans cet exemple, seules les capacités de HBA des adresses de Hôte1 vont été utilisées. En d'autres termes, ni la clé publique incluse dans la structure de données de paramètre de CGA ni sa clé privée correspondante n'ont été utilisées dans le protocole. Dans le paragraphe suivant, on va considérer un cas où leur usage est nécessaire.

8.1. Prise en charge d'ensemble d'adresses dynamique

Dans le paragraphe précédent, on a présenté les mécanismes qui permettent à un hôte d'utiliser différentes adresses d'un ensemble prédéterminé, pour échanger les paquets d'une communication. L'ensemble d'adresses impliqué a été prédéterminé et est connu quand la communication est initiée. Pour réaliser une telle fonction, seules les fonctions de HBA des adresses étaient nécessaires. Dans ce paragraphe, on va explorer le cas où le but est d'échanger des paquets en utilisant des adresses supplémentaires qui n'étaient pas connues à l'établissement de la communication. Un exemple de cette situation est quand un nouveau préfixe est disponible dans un site après un événement de renumérotation. Dans ce cas, les hôtes qui ont la nouvelle adresse disponible peut vouloir l'utiliser dans les communications qui ont été établies avant l'événement de renumérotage. Dans ce cas, les fonctions HBA des adresses ne sont pas suffisantes et les capacités de CGA sont à utiliser.

Considérons ensuite le cas de communication précédent entre Hôte1 et Hôte2. Supposons que la communication est active et en cours, comme décrit précédemment. Hôte1 utilise PA:LA:iidA et Hôte2 utilise IPHôte2 pour échanger des paquets. Maintenant supposons qu'une nouvelle adresse, PC:LC: addC soit disponible chez Hôte1. Noter que cette adresse est juste une adresse IPv6 régulière, et n'est ni une HBA ni une CGA. Hôte1 veut utiliser cette nouvelle adresse dans la communication existante avec Hôte2. On devrait noter que le mécanisme de HBA décrit au paragraphe précédent ne peut pas être utilisé pour vérifier cette nouvelle adresse, car elle n'appartient pas à l'ensemble de HBA (car le préfixe n'était pas disponible au moment de la génération de l'ensemble de HBA). Cela signifie que des mécanismes de vérification supplémentaires vont être nécessaires.

Afin de vérifier cette nouvelle adresse, les capacités de CGA de PA:LA:iidA sont utilisées. Noter que la même adresse est utilisée, seul le mécanisme de vérification est différent. Donc, si Hôte1 veut utiliser PC: LC:addC pour échanger des paquets dans la communication établie, il va utiliser le message UPDATE défini dans le protocole Shim6 [RFC5533], portant la nouvelle adresse, PC:LC:addC, et ce message va être signé en utilisant la clé privée correspondant à la clé publique contenue dans CGA_PDS_A. Quand Hôte2 reçoit le message, il vérifie la signature en utilisant la clé publique contenue dans la structure de données de paramètre de CGA associée à l'adresse utilisée pour établir la communication, c'est-à-dire, CGA_PDS_A et PA:LA:iidA, respectivement. Une fois la signature vérifiée, la nouvelle adresse (PC:LC:addC) peut être utilisée dans la communication.

Dans tous les cas, un événement de renumérotage a un impact sur un site qui utilise la technique de HBA. En particulier, le nouveau préfixe ajouté ne va pas être inclus dans l'ensemble de HBA existant, donc il est seulement possible d'utiliser le nouveau préfixe avec l'ensemble existant de HBA si les capacités de CGA sont utilisées. Bien que ce soit acceptable à court terme, à long terme, le site va devoir renuméroter ses adresses HBA. Pour le faire, il va devoir re-générer les ensembles de HBA alloués aux hôtes en incluant le nouveau préfixe dans l'ensemble de préfixes, ce qui va résulter en des adresses différentes, non seulement parce que on a besoin d'ajouter une nouvelle adresse avec le nouveau préfixe, mais aussi parce que les adresses avec les préfixes existants vont aussi changer à cause de l'inclusion d'un nouveau préfixe dans l'ensemble de préfixes. De plus, comme les adresses HBA doivent être générées localement, une fois que celles-ci sont générées après l'événement de renumérotation, les nouvelles informations d'adresses doivent être portées au gestionnaire DNS au cas où ces informations d'adresse seraient à publier dans le DNS (voir la section des considérations sur le DNS pour les détails).

9. Considérations sur le DNS

Les ensembles de HBA peuvent être générés en utilisant tout ensemble de préfixes. En fait, la seule particularité des HBA est qu'elles contiennent des informations sur l'ensemble de préfixes dans la partie identifiant d'interface de l'adresse sous la forme d'un hachage, mais aucune hypothèse n'est faite sur les propriétés des préfixes utilisés pour la génération de HBA. Cela signifie que selon les préfixes utilisés pour la génération d'ensemble de HBA, il peut ou non être recommandé de publier les adresses résultantes (HBA) dans le DNS. Par exemple, quand des préfixes d'adresse locale unique (ULA, *Unique Local Address*) [RFC4193] sont inclus dans le processus de génération de HBA, des considérations spécifiques du DNS relatives à la nature locale de l'ULA devrait être prises en compte et les recommandations appropriées relatives à la publication de tels préfixes dans le DNS devraient être suivies. De plus, parmi ses adresses, un hôte donné peut avoir des HBA et d'autres adresses IPv6. La conséquence de cela est que seules les adresses HBA vont être liées ensemble par la technique de HBA, tandis que les autres adresses ne seront pas liées à l'ensemble de HBA. Cela signifierait fondamentalement que si une des autres adresses est utilisée pour initier une communication Shim6, il ne va pas être possible d'utiliser la technique de HBA pour lier l'adresse utilisée à l'ensemble de HBA. De plus, comme les adresses HBA ne sont pas distinguables des autres adresses IPv6 dans leur format, un initiateur ne va pas être capable de distinguer, en regardant simplement les différentes adresses, lesquelles appartiennent à l'ensemble de HBA et celles qui ne le font pas, donc des moyens supplémentaires vont être nécessaires quand l'initiateur est supposé utiliser seulement des HBA pour établir des communications en présence d'adresses non HBA dans le DNS.

De plus, on devrait noter que les valeurs réelles de HBA sont le résultat d'une procédure de génération de HBA, ce qui signifie qu'elles ne peuvent pas être choisies arbitrairement. Cela a une implication à l'égard de la gestion du DNS, parce que la partie qui génère l'ensemble d'adresses HBA doit porter les informations d'adresse au gestionnaire DNS, afin que les adresses soient publiées et pas autrement. La situation est similaire à celle des adresses CGA régulières et même au cas où l'auto-configuration d'adresse sans état est utilisée. Pour faire cela, il est possible d'utiliser les mises à jour dynamiques du DNS [RFC2136] ou d'autres outils propriétaires. Une considération similaire s'applique quand l'hôte veut publier des entrées inverse du DNS. Comme l'hôte a besoin de générer ses adresses HBA, il va devoir porter les informations d'adresse au gestionnaire du DNS afin que l'entrée inverse du DNS appropriée soit remplie au cas où c'est nécessaire. On devrait noter que ni le protocole Shim6 ni la technique de HBA ne s'appuient sur le DNS inverse pour leur fonctionnement approprié et les raisons générales pour exiger le remplissage du DNS inverse s'appliquent comme pour toute autre adresse IPv6 régulière.

10. Considérations relatives à l'IANA

Le présent document définit une nouvelle extension de CGA, l'extension multi préfixes. Cette extension a reçu la valeur de type d'extension CGA de 0x0012.

11. Considérations sur la sécurité

Le but des HBA est de créer un groupe d'adresses liées de façon sûre, afin qu'elles puissent être utilisées de façon interchangeable quand on communique avec un nœud. Si il n'y a pas de lien sûr entre les différentes adresses d'un nœud, un certain nombre d'attaques sont possibles, comme décrit dans la [RFC4218]. En particulier, il serait possible à un attaquant de rediriger les communications d'une victime sur une adresse choisie par l'attaquant, capturant la communication. Quand on utilise des HBA, seules les adresses appartenant à un ensemble de HBA peuvent être utilisées de façon interchangeable, limitant les adresses qui peuvent être utilisées pour rediriger la communication à un ensemble prédéterminé qui appartient au nœud original impliqué dans la communication. Donc, quand il utilise des HBA, un nœud qui communique en utilisant l'adresse A peut rediriger la communication sur une nouvelle adresse B si et seulement si B appartient au même ensemble de HBA que A.

Cela signifie que si un attaquant veut rediriger les communications adressées à l'adresse HBA1 sur une adresse de remplacement IPX, l'attaquant va devoir créer une structure de données de paramètre de CGA qui génère un ensemble de HBA qui contienne à la fois HBA1 et IPX.

Afin de générer l'ensemble de HBA requis, l'attaquant devra trouver une structure de données de paramètre de CGA qui satisfasse les conditions suivantes :
- o le préfixe de HBA1 et le préfixe de IPX sont inclus dans l'extension multi préfixes,
- o HBA1 est inclus dans l'ensemble de HBA généré.

Note : cela suppose qu'il soit acceptable pour l'attaquant de rediriger HBA1 sur toute adresse du préfixe de IPX.

Les champs restants qui peuvent être changés à la discrétion de l'attaquant afin de satisfaire les conditions ci-dessus sont : le modificateur, les autres préfixes dans l'extension multi préfixes, et les autres extensions. Dans tous les cas, afin d'obtenir l'ensemble de HBA désiré, l'attaquant va devoir utiliser une attaque en force brute, ce qui implique de générer de multiples ensembles de HBA avec des paramètres différents (par exemple avec un modificateur différent) jusqu'à ce que les conditions désirées soient satisfaites. Le nombre de fois attendu que le processus de génération va devoir être répété jusqu'à ce que l'ensemble de HBA désiré soit trouvé est en rapport exponentiel avec le nombre de bits contenant les informations de hachage incluses dans l'identifiant d'interface de la HBA. Comme 59 des 64 bits de l'identifiant d'interface contiennent des bits de hachage, le nombre de générations attendues qui va devoir être effectué par l'attaquant est O(2^59). Note : on suppose que la force brute est la meilleure attaque contre les HBA/CGA. Aussi, noter que l'hypothèse que l'outil Sec défini dans la [RFC3972] multiplie le facteur d'attaque tient pour les attaques en force brute mais ne peut pas tenir pour les autres classes d'attaque.

La protection contre les attaques en force brute peut être améliorée en augmentant le paramètre Sec. Un paramètre Sec non zéro implique que les étapes 3-4 du processus de génération vont être répétées O(2^(16*Sec)) fois (nombre de fois attendu). Si on assimile le coût de répéter les étapes 3-4 au coût de génération de l' adresse HBA, on peut estimer le nombre de fois que la génération est à répéter à O(2^(59+16*Sec)) dans le cas de valeurs de Sec de 1 et 2. Pour d'autres valeurs de Sec, les mécanismes de protection de Sec seront définies par les spécifications mentionnées par le registre CGA SEC défini dans la

[RFC4982].

11.1 Considérations sur la sécurité quand on utilise des HBA dans le protocole Shim6

Dans ce paragraphe, on analyse la sécurité fournie par les HBA dans le contexte d'un protocole Shim6 comme décrit à la Section 8 de ce mémoire.

Tout d'abord, on doit noter que les HBA ne peuvent pas empêcher des attaques par interposition (MITM). Cela signifie que dans le scénario décrit à la Section 8, si un attaquant est situé le long du chemin entre Hôte1 et Hôte2 pendant la durée de vie de la communication, l'attaquant va être capable de changer les adresses utilisées pour la communication. Cela signifie que il va être capable de changer les adresses utilisées dans la communication, d'ajouter ou supprimer des préfixes à sa guise. Cependant, l'attaquant doit s'assurer que la structure de données de paramètre de CGA et l'ensemble de HBA sont changés en conséquence. Cela signifie essentiellement que l'attaquant va devoir changer la partie identifiant d'interface des adresses impliquées, car un changement dans l'ensemble de préfixes va résulter en des identifiants d'interface différents des adresses de l'ensemble de HBA, sauf si la valeur appropriée de modificateur est utilisée (ce qui exigerait O(2(59+16*Sec)) tentatives). Donc, HBA ne fournit pas protection des attaques de MITM, mais un attaquant interposé va devoir changer l'adresse utilisée dans la communication afin de changer l'ensemble de préfixes valide pour la communication.

Les HBA fournissent une protection contre les attaques en temps glissant [RFC4218], [RFC4225]. Dans le contexte de multi rattachements, un attaquant va effectuer une attaque en temps glissant de la façon suivante : un attaquant placé le long du chemin de la communication va modifier les paquets pour inclure une adresse supplémentaire comme adresse valide pour la communication. Ensuite l'attaquant va quitter la localisation sur le chemin, mais les effets de l'attaque vont rester (c'est-à-dire, l'adresse va encore être considérée comme une adresse valide pour cette communication). On va ensuite présenter comment les HBA peuvent être utilisées pour empêcher de telles attaques.

Si l'attaquant n'est pas sur le chemin quand la structure de données de paramètre de CGA initiale est échangée, sa seule possibilité de lancer une attaque de redirection est de falsifier la signature du message pour ajouter de nouvelles adresses en utilisant les capacités de CGA des adresses. Cela implique de découvrir la clé publique utilisée dans la structure de données de paramètre de CGA et ensuite de casser la paire de clés, ce qui ne semble pas faisable. Donc, afin de lancer une attaque de redirection, l'attaquant doit être dans le chemin quand la structure de données de paramètre de CGA est échangée, et pouvoir la modifier. Maintenant, afin de lancer l'attaque de redirection, l'attaquant doit ajouter son propre préfixe dans l'ensemble de préfixes de la structure de données de paramètre de CGA. On a vu au paragraphe précédent qu'il y a deux approches possibles pour cela :

1. Trouver la bonne valeur de modificateur, afin que l'adresse initialement utilisée dans la communication soit contenue dans le nouvel ensemble de HBA. Le coût de cette attaque est O(2(59+16*Sec)) itérations du processus de génération, donc elle est réputée infaisable.

2. Utiliser une valeur quelconque de modificateur, afin que l'adresse initialement utilisée dans la communication soit probablement non incluse dans l'ensemble de HBA. Dans ce cas, l'attaquant doit rester en chemin, car il a besoin de réécrire l'adresse portée dans les paquets (sinon, les points d'extrémité vont remarquer un changement de l'adresse utilisée dans la communication). Cela signifie essentiellement que l'attaquant ne peut pas lancer une attaque en temps glissant mais doit être un interposé à plein temps.

Donc, la conclusion est que les HBA fournissent la protection contre les attaques en temps glissant.

Les HBA ne fournissent pas une protection complète contre les attaques d'inondation, et, par suite, le protocole SHIM6 a d'autres moyens de les traiter. Cependant, les HBA rendent très difficile de lancer une attaque d'inondation contre une adresse spécifique. Il est possible cependant de lancer une attaque d'inondation contre un préfixe. Et bien sûr, la protection qu'offrent les HBA s'applique seulement aux nœuds qui l'emploient ; HBA ne fournit pas de solution pour la protection générale contre les attaques d'inondation pour les autres nœuds.

Supposons qu'un attaquant ait un accès aisé à un préfixe PX::/nX et qu'il veuille lancer une attaque d'inondation sur un hôte situé à l'adresse P:iid. L'attaque va consister à établir une communication avec un serveur S et à lui demander un fort flux. Ensuite en le redirigeant simplement sur P:iid, d'inonder la cible. Pour effectuer cette attaque, l'attaquant doit générer un ensemble de HBA incluant P et PX dans l'ensemble de préfixes, et être sûr que l'ensemble de HBA résultant contient P:iid. Pour faire cela, l'attaquant doit trouver la valeur appropriée de modificateur. Le nombre de tentatives attendues requis pour trouver une telle valeur de modificateur est O(2(59+16*Sec)), comme présenté plus haut. Donc, on peut conclure qu'une telle attaque n'est pas faisable.

Cependant, la cible d'une attaque d'inondation n'est pas limitée à des hôtes spécifiques, mais elle peut aussi être lancée contre d'autres éléments de l'infrastructure, comme un routeur ou des liaisons d'accès. Pour faire cela, l'attaquant peut établir une communication avec un serveur S et demander le téléchargement d'un gros flux. Ensuite, l'attaquant redirige la communication sur une adresse du réseau cible. Même si l'adresse cible n'est pas allouée à un hôte, le flux va inonder la liaison d'accès du cite cible, et le routeur d'accès au site va aussi souffrir de la surcharge. De telles attaques ne peuvent pas être empêchées en utilisant les HBA, car l'attaquant peut facilement générer un ensemble de HBA en utilisant son propre préfixe et le préfixe du réseau cible. Pour empêcher de telles attaques, des mécanismes supplémentaires sont nécessaires, comme des essais d'accessibilité.

11.2 Considérations de confidentialité

Les HBA peuvent être utilisées comme des adresses de la [RFC4941]. Si un nœud veut utiliser des adresses temporaires, il va devoir générer périodiquement de nouveaux ensembles de HBA. L'effort requis pour cette opération dépend de la valeur du paramètre Sec. Si Sec=0, alors le coût de génération d'un nouvel ensemble de HBA est similaire au coût de génération d'un nombre aléatoire, c'est-à-dire, une itération de la procédure de génération de l'ensemble de HBA. Cependant, si Sec>0, alors le coût de génération d'un ensemble de HBA est significativement augmenté, car il faut O(2(16*Sec)) itérations du processus de génération. Dans ce cas, selon la fréquence de changement d'adresse requis, la prise en charge de l'adresse RFC 4941 peut être plus coûteuse.

11.3 Considérations de dépendance à SHA-1

De récentes attaques sur des fonctions de hachage couramment utilisées ont motivé beaucoup de soucis dans la communauté de l'Internet. L'approche recommandée [RFC4270], [15] pour traiter ce problème est d'abord d'analyser l'impact de ces attaques sur les différents protocoles Internet qui utilisent les fonctions de hachage, et ensuite de s'assurer que les différents protocoles Internet qui utilisent ces fonctions de hachage sont capables de migrer sur une fonction de hachage de remplacement (plus sûre) sans perturbations majeures du fonctionnement de l'Internet.

L'analyse susmentionnée pour les CGA et leurs extensions (y compris les HBA) est effectuée dans la [RFC4982]. La conclusion de l'analyse est que la sécurité des protocoles qui utilisent les CGA et leurs extensions n'est pas affectée par les attaques récemment disponibles contre les fonctions de hachage. En dépit de cela, la spécification de CGA [RFC3972] a été mise à jour par la [RFC4982] pour permettre la prise en charge de fonctions de hachage de remplacement.

11.4 Considérations des attaques de DoS

Pour utiliser la technique de la HBA, le propriétaire de l'ensemble de HBA doit informer son homologue de la structure de données de paramètre de CGA afin de lui permettre de vérifier que les différentes HBA appartiennent au même ensemble de HBA. Ces informations doivent alors être mémorisées par l'homologue pour vérifier à l'avenir les adresses de remplacement. Cela peut être un vecteur pour des attaques de DoS, car l'homologue doit engager des ressources (dans ce cas particulier, de la mémoire) pour être capable d'utiliser la technique de HBA pour la vérification d'adresse. Il est alors possible à un attaquant de lancer une attaque de DoS en apportant des informations de HBA à une victime, lui imposant d'utiliser de la mémoire pour mémoriser l'état relatif à la HBA, et épuisant éventuellement la mémoire pour d'autres opérations authentiques. Pour empêcher une telle attaque, les protocoles qui utilisent la technique de HBA devraient mettre en œuvre des techniques de prévention de DoS appropriées.

Par exemple, le protocole Shim6 [RFC5533] inclut une prise de contact en quatre phases pour établir le contexte Shim6 et, en particulier, pour établir l'état relatif à HBA. Dans cette prise de contact en quatre phases, le receveur reste sans état durant les deux premiers messages, tandis que l'initiateur doit garder l'état pendant tout l'échange des quatre messages afin que le coût de l'établissement de contexte soit plus élevé en termes de mémoire pour l'initiateur (c'est-à-dire, le potentiel attaquant) que pour le receveur (c'est-à-dire, la victime potentielle). En plus de cela, la prise de contact en quatre phases empêche l'usage d'adresses falsifiées provenant d'un attaquant hors du chemin, car l'initiateur doit être capable de recevoir des informations sur l'adresse qu'il a utilisé comme adresse de source, permettant le traçage de la localisation d'où l'attaque est lancée.

12. Contributeurs

Le présent document a été à l'origine produit par une équipe de conception MULTI6 constituée de (en ordre alphabétique) : Jari Arkko, Marcelo Bagnulo, Iljitsch van Beijnum, Geoff Huston, Erik Nordmark, Margaret Wasserman, et Jukka Ylitalo.

13. Remerciements

La discussion initiale sur HBA a bénéficié de contributions de Alberto Garcia-Martinez, Tuomas Aura, et Arturo Azcorra.

Les processus de génération d'ensemble de HBA et de vérification de HBA décrits dans le présent document contiennent plusieurs étapes extraites de la [RFC3972].

Jari Arkko, Matthew Ford, Francis Dupont, Mohan Parthasarathy, Pekka Savola, Brian Carpenter, Eric Rescorla, Robin Whittle, Matthijs Mekking, Hannes Tschofenig, Spencer Dawkins, Lars Eggert, Tim Polk, Peter Koch, Niclas Comstedt, David Ward, et Sam Hartman ont relu ce document et fourni de précieux commentaires.

Le texte inclus dans le paragraphe 3.2 a été fourni par Eric Rescorla.

L'auteur tient aussi à remercier Francis Dupont pour la fourniture de la première mise en œuvre de HBA.

14. Références

14.1 Références normatives

[RFC2119] S. Bradner, "Mots clés à utiliser dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. *(MàJ par* RFC8174)

[RFC3279] L. Bassham, W. Polk et R. Housley, "Algorithmes et identifiants pour le profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", avril 2002.

[RFC3971] J. Arkko et autres, "Découverte de voisin sûre (SEND)", mars 2005. (*MàJ par* RFC6494) *(P.S.)*

[RFC3972] T. Aura, "Adresses générées cryptographiquement (CGA)", mars 2005. (*MàJ par* RFC4581, RFC4982) *(P.S.)*

[RFC4291] R. Hinden, S. Deering, "Architecture d'adressage IP version 6", février 2006, DOI 10.17487/RFC4291. (*MàJ par* 5952 et 6052, 8064) *(D.S.)*

[RFC4581] M. Bagnulo, J. Arkko, "Format de champ d'extension des adresses générées cryptographiquement (CGA)", octobre 2006. (*MàJ* RFC3972) *(P.S.)*

[RFC4941] T. Narten et autres, "Extensions de confidentialité pour l'auto configuration d'adresse sans état dans IPv6", septembre 2007, DOI 10.17487/RFC4941. (*D.S. ; remplace* RFC3041 ; r*emplacée par* RFC8981*)*

[RFC4982] M. Bagnulo, J. Arkko, "Prise en charge de plusieurs algorithmes de hachage dans les adresses générées cryptographiquement (CGA)", juillet 2007. (*MàJ* RFC3972) *(P.S.)*

[RFC5280] D. Cooper et autres, "Profil de certificat d'infrastructure de clé publique X.509 et de liste de révocation de certificat (CRL) pour l'Internet", mai 2008. (*Remplace les* RFC3280, RFC4325, RFC4630) *(P.S. ; MàJ par* RFC8398, 8399)

[RFC5533] E. Nordmark, M. Bagnulo, "Shim6 : Protocole Shim de niveau 3 de multi rattachement pour IPv6", juin 2009. *(P. S.)*

14.2 Références pour information

[RFC2136] P. Vixie, S. Thomson, Y. Rekhter et J. Bound, "Mises à jour dynamiques dans le système de noms de domaine (DNS UPDATE)", avril 1997, DOI 10.17487/RFC2136.

[RFC4193] R. Hinden, B. Haberman, "Adresses IPv6 en envoi individuel uniques localement", octobre 2005. *(P.S.)*

[RFC4218] E. Nordmark, T. Li, "Menaces qui pèsent sur les solutions de rattachement multiple IPv6", octobre 2005. *(Information)*

[RFC4225] P. Nikander et autres, "Fondements des concepts de sécurité de l'optimisation de l'acheminement d'IPv6 mobile", décembre 2005. *(Information)*

[RFC4270] P. Hoffman, B. Schneier, "Attaques contre les hachages cryptographiques dans les protocoles Internet", nov. 2005. *(Info.)*

[RFC4866] J. Arkko, C. Vogt, W. Haddad, "Optimisation d'acheminement amélioré pour IPv6 mobile", mai 2007. *(P.S.)*

[15] Bellovin, S. and E. Rescorla, "Deploying a New Hash Algorithm", septembre 2005.

[16] Nordmark, E., "Multi6 Application Referral Issues", Travail en cours, octobre 2004.

[17] Bagnulo, M., Garcia-Martinez, A., and A. Azcorra, "Efficient Security for IPv6 Multihoming", ACM Computer Communications Review Vol 35 n 2, avril 2005.

Adresse de l'auteur

Marcelo Bagnulo
Universidad Carlos III de Madrid
Av. Universidad 30
Leganes, Madrid 28911
Espagne
téléphone : 34 91 6249500
mél : marcelo@it.uc3m.es
URI : http://www.it.uc3m.es