Identification de flux IPv6 par étalement des adresses

Florent Fourcot (florent.fourcot@telecom-bretagne.eu)[†*]
Laurent Toutain (laurent.toutain@telecom-bretagne.eu)[*]
Frédéric Cuppens (frederic.cuppens@telecom-bretagne.eu)[*]
Nora Cuppens-Boulahia (nora.cuppens@telecom-bretagne.eu)[*]
Stefan Köpsell (stefan.koepsell@tu-dresden.de)[†]

Résumé : L'identification des flux réseaux est une fonctionnalité très importante pour assurer la sécurité d'Internet. Cette identification est habituellement effectuée grâce à cinq identifiants bien connus : les adresses IP sources et destinations, le numéro de protocole de la couche transport, et les deux identifiants de la couche transport (appelés ports en UDP et TCP). Malheureusement, les identifiants de la source ne sont absolument pas fiables.
Avec l'introduction de l'IPv6, il est possible de bâtir de nouveaux paradigmes de sécurité basés sur les nouveautés de ce protocole. En particulier, IPv6 introduit un immense espace d'adressage. Nous proposons d'en tirer parti en changeant très fréquemment les adresses d'un flux de données, mécanisme que nous appelons l'étalement d'adresse.
Cet étalement améliore l'identification des flux, car seuls les équipements source et destination connaissent la série d'adresses utilisée. Un attaquant sera ainsi incapable d'introduire un paquet à l'intérieur d'un flux existant ou d'initialiser un flux.

Mots Clés : IPv6, sécurité, identification de flux

1 Introduction

1.1 Identification des flux sur Internet

L'identification des flux est à la base de certaines fonctionnalités d'Internet, comme la sécurité (filtrage des paquets) et les politiques de hiérarchisation des priorités pour la qualité de service. Cependant, le protocole IP à la base d'Internet est construit sur un réseau à datagrammes, les paquets sont indépendants et peuvent suivre différentes routes. La notion de flux n'existe pas au niveau de la couche réseau. Un flux est donc défini par la RFC 2722 [BMR99] comme un « équivalent artificiel logique à un appel ou à une connexion ». C'est artificiel, car il n'existe pas de moyens simples permettant de séparer les paquets en flux dans un réseau IP.

La notion de flux est plus naturelle pour la couche transport. Elle est en effet indispensable pour trier les paquets, rassembler les segments et détecter les erreurs, comme peut le faire TCP. Pour obtenir cette notion de flux, on peut donc utiliser les identifiants de la couche transport, appelés ports pour TCP.

∗. Institut Mines-Télécom; Télécom Bretagne
†. TU Dresden; Faculty of Computer Science

Pour identifier un flux, il est nécessaire d'extraire un n-uplet composé de cinq éléments. Les deux premiers sont les adresses sources et destinations, lisibles directement dans l'entête IP. Le suivant est le numéro du protocole de la couche transport, disponible dans le champ `next header` pour un paquet IPv6 sans extension. Avec l'aide de ce numéro, on peut analyser les entêtes de la couche transport pour lire les numéros de ports source et destination, qui complètent le n-uplet.

Cette méthode bien connue est la base de l'identification d'un flux. Cette identification peut-être plus complexe, par exemple un pare-feu avec état doit suivre et sauvegarder les états des connexions TCP, afin de rejeter les paquets ne suivant pas le standard TCP.

1.2 Avantages de l'étalement d'adresse

Les cinq éléments du n-uplet permettant l'identification d'un flux ne sont pas fiables par nature. Notamment, l'adresse IP source et le port source peuvent être très facilement manipulés par un attaquant. Si un attaquant est en mesure d'envoyer un paquet avec une adresse source falsifiée, il sera en mesure de réinitialiser une connexion TCP, d'injecter des paquets sur le réseau, d'effectuer une attaque ciblée sur la destination, etc.

Avec IPv6, l'immense espace d'adressage apporte de nouvelles opportunités pour la sécurité, comme les adresses générés cryptographiquement [Aur05]. IPv6 permet d'utiliser un grand nombre d'adresses, contrairement aux réseaux IPv4 qui sont obligés de minimiser ce nombre. Notre solution améliore la sécurité par l'étalement d'adresses, c'est à dire par le renouvellement très fréquent des adresses sources et destinations, en suivant une séquence temporelle. Cette séquence n'est connue que par les équipements en communication, améliorant ainsi la robustesse de l'identification.

Comme nous ne modifions que les adresses IP, notre solution est simple. Elle ne nécessite pas d'encapsulation (comme un tunnel IPsec le fait) et peut-être suivie par un pare-feu adapté connaissant un secret partagé.

1.3 Modèle de l'attaquant

Notre attaquant a la capacité d'injecter des paquets avec des adresses sources arbitraires. Il peut-être situé sur le chemin de la transmission et lire le trafic légitime.

Nous n'essayons pas de protéger contre la reconstruction d'un flux, un attaquant peut donc utiliser les informations des couches supérieures pour identifier la série de paquets d'un flux. Notre objectif est de protéger contre l'usurpation : nos étaleurs reconnaissent les paquets forgés par un attaquant.

Cependant, notre étalement camoufle les adresses IP et protège donc contre la corrélation des flux. Un attaquant ne peut pas deviner l'adresse IP source réelle d'un flux, et ne peut donc pas relier plusieurs flux à une source particulière à l'aide des entêtes IP.

1.4 Contrainte de l'étalement : les deux parties d'une adresse IPv6

Les adresses IPv6 [1] sont divisées en deux parties. La première, habituellement appelée « préfixe », est nécessaire au routage. Elle n'est donc pas modifiable, car un routeur ne pourra plus transmettre correctement un paquet dont le préfixe de destination est modifié. De la même manière, l'adresse IP source doit respecter les règles contre l'usurpation spécifiées dans la RFC 2827 [FS00] et ne peut pas être réécrite arbitrairement.

1. Composées de 128 bits, contre 32 bits en IPv4

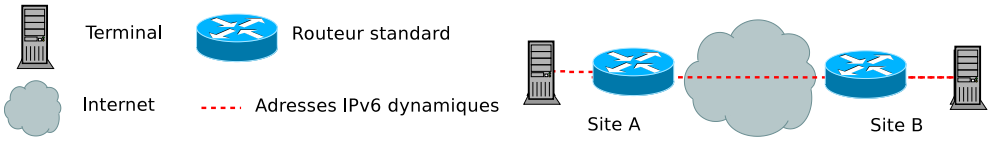

Figure 1: Étalement d'adresse sans équipement supplémentaire

Si aucune recommandation sur la taille du préfixe n'est précisée, sa longueur ne doit pas être supérieure à 64 bits pour la compatibilité avec l'auto-configuration IPv6. En effet, l'auto-configuration sans état des adresses IPv6 dérive la seconde partie d'une adresse (que l'on appelle identifiant d'interface) de l'adresse MAC du matériel en un identifiant à 64 bits.

Nous avons donc choisi d'étaler les derniers 64 bits de chaque adresse seulement. Cette valeur devrait être compatible avec la plupart des réseaux. Cette architecture peut être utilisée avec la délégation de préfixe, où chaque terminal est son propre étaleur.

1.5 Plan de l'article

Nous présentons tout d'abord quelques problèmes liés à l'étalement d'adresses et l'architecture de notre solution en section 2. Nous introduisons ensuite les principes de l'étalement et nos notations en section 3. Nous décrivons pas à pas l'initialisation d'une connexion en section 4, que nous complétons par le traitement d'un paquet par un étaleur en section 5.

2 Emplacement de l'étaleur

2.1 Contraintes de l'étalement sur un équipement terminal

L'idée naturelle de l'étalement est de générer et de suivre les séquences d'adresses sur les équipements terminaux, comme sur la figure 1. Cette architecture permet une sécurité de bout en bout, il n'est pas nécessaire de déléguer la sécurité à d'autres équipements.

Cependant, cette solution implique une mise à jour du routeur si un préfixe individuel n'est pas délégué à chaque équipement. En effet, dans le cas d'un partage de préfixe entre plusieurs équipements, l'utilisation de nombreuses adresses provoquera :

— une inondation du réseau avec des paquets de découvertes de voisins. Le routeur n'ayant pas connaissance de l'étalement, il ne peut pas établir le lien entre la série d'adresses IP et l'adresse MAC ;
— une saturation de la table des voisins du routeur. Avec des changements fréquents d'adresses, le routeur ne sera pas en capacité de sauvegarder tous les liens entre adresses IP et adresses MAC ;
— une augmentation de la latence à chaque changement d'adresse IP, causé par la découverte de voisins.

Pour résoudre ce problème, une solution est de mettre à jour le routeur pour suivre une séquence d'adresse IP dans la table des voisins. Le routeur n'a pas besoin de connaître les deux séries d'adresses (source et destination), mais uniquement la séquence des adresses sources reçues d'Internet. Il ne sera pas en mesure d'insérer un paquet dans un flux, l'information reste partielle.

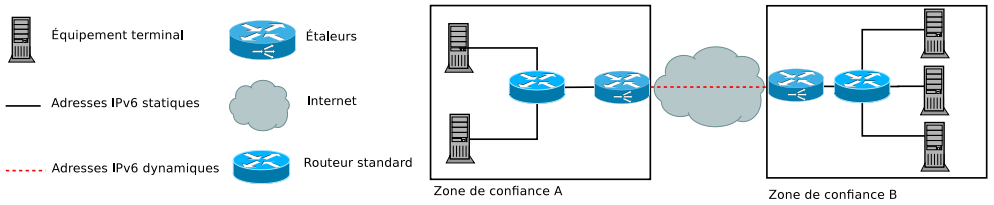

Figure 2: Architecture de la solution : étaleurs en bordure de la zone de confiance

2.2 La délégation de préfixe

Une autre approche serait de déléguer un préfixe à chaque équipement, solution réaliste qui ne provoquera pas une pénurie d'adresse sur un réseau IPv6. Cela résout le problème du lien entre les adresses MAC et les adresses IP, car les intermédiaires enverraient les paquets correspondants à un préfixe sans se soucier de l'identifiant d'interface.

Avec une telle architecture, l'équipement terminal est en charge de la gestion des identifiants d'interface et peut être vu comme un routeur. Cette délégation de préfixes est la meilleure architecture pour la simplicité de la solution et du point de vue de la sécurité. Cependant, elle ne serait pas compatible avec tous les réseaux.

2.3 Simplification de l'architecture avec l'introduction d'étaleurs

Pour une première approche de l'étalement, nous proposons donc de simplifier l'architecture en ajoutant de nouveaux équipements nommés « étaleurs ». Ces équipements sont capables de réécrire un flux de paquets à adresses statiques en un flux de paquets utilisant des adresses dynamiques. L'étaleur peut-être directement en bordure du réseau ou en bordure d'une zone de confiance (voir la figure 2).

Cette architecture simplifie le déploiement de la solution. Un administrateur n'a plus besoin de mettre à jour et de configurer chaque équipement, mais peut simplement insérer l'étaleur dans le réseau. Un second avantage est la possibilité de filtrage des paquets. Les paquets malveillants utilisent des ressources, et peuvent saturer la capacité du réseau. Il est donc utile de filtrer les paquets malveillants aussi tôt que possible, et non pas au niveau de l'équipement terminal.

Nous avons donc choisi cette architecture pour cet article afin de simplifier les concepts et les expérimentations.

3 Principes généraux

3.1 Prérequis de la solution

Pour activer l'étalement, il est nécessaire de configurer au moins deux réseaux, capables d'effectuer le lien entre les adresses dynamiques et les identifiants stables. Cette configuration est réalisée en ajoutant un étaleur en bordure de chaque réseau. Ces étaleurs partagent un secret, qu'un attaquant ne peut deviner. La communication de ce secret est hors du cadre de cette publication.

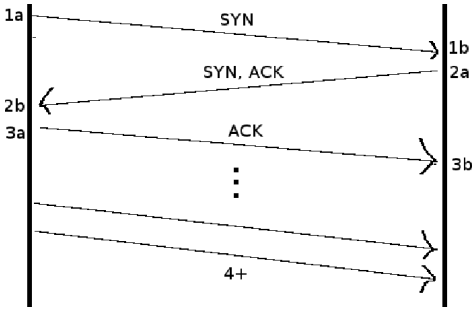

Figure 3: Chronogramme de l'initialisation d'une connexion

3.2 Initialisation d'un étaleur

L'initialisation d'un étaleur consiste à créer une configuration pour chaque réseau pair compatible. La configuration contient la liste de préfixes de la destination et une fonction pour dériver une clef cryptographique à partir du secret partagé.

3.3 Échange des données de sessions

Un de nos objectif est d'étaler chaque flux en une unique séquence d'adresses. Les deux étaleurs doivent donc échanger des données de session à chaque initialisation d'un flux. Cette transmission d'information peut se faire de plusieurs manières.

La première d'entre elle est d'échanger des paquets supplémentaires pour initialiser le contexte de chaque flux. Cela augmente la latence à l'initialisation, et coûte des ressources.

La seconde est d'ajouter des informations réelles dans un paquet réel, en ajoutant par exemple une extension à l'entête IPv6. Comme cette extension est ajoutée par l'étaleur et non par le terminal, cela peut conduire à un dépassement de la taille maximale d'un paquet. Comme il est interdit de fragmenter les paquets IPv6 en transit, il est nécessaire de transmettre un paquet ICMP à l'émetteur, pour réduire la taille des paquets envoyés. Cela réduira les performances pour toute la session.

Ces solutions ne sont donc pas satisfaisantes, et nous en proposons une autre. Nos étaleurs encodent toutes les informations de session dans les adresses IPv6, et n'ajoutent aucune entête ou paquet supplémentaire. Cela limite la quantité d'information échangeables, mais n'a aucun coût en terme de ressource ou de latence.

3.4 Notations

Nous introduisons les notations suivantes pour la réécriture d'adresses : P_A et P_B sont les préfixes des réseaux hôtes A et B. IID_A et IID_B sont les identifiants d'interfaces des hôtes A et B. Ces valeurs sont concaténées en IP_A et IP_B, adresses IP réelles des hôtes.

IP^n_{src} est l'adresse IP source réécrite en étape n, et IP^n_{dst} l'adresse IP destination réécrite à la même étape. Comme il est impossible de réécrire les préfixes, IP^n_{src} et IP^n_{dst} sont la concaténation des préfixes stables (P_A ou P_B) et d'une valeur réécrite.

Le tableau 1 résume ces notations en fonction des étapes. La description de notre protocole suit les mêmes étapes qu'une initialisation TCP (voir la figure 3).

	Réseau local		Internet	
Étape	IP source réelle	IP destination réelle	IP source réécrite	IP destination réécrite
1a → *1b* SYN	IP_A	IP_B	$IP^1_{src} = P_A \vert IID^1_{src}$	$IP^1_{dst} = P_B \vert IID^1_{dst}$
2a → *2b* SYN, ACK	IP_B	IP_A	$IP^2_{src} = P_B \vert IID^2_{src}$	$IP^2_{dst} = P_A \vert IID^2_{dst}$
3a → *3b* ACK	IP_A	IP_B	$IP^3_{src} = P_A \vert IID^3_{src}$	$IP^3_{dst} = P_B \vert IID^3_{dst}$

Table 1: Notation pour les adresses IP

4 Description pas à pas de l'initialisation

4.1 Initialisation d'une connexion : premier paquet

4.1.1 Réécriture symétrique sur les étaleurs

L'étalement commence à l'étape *1a*, lorsque l'étaleur reçoit le premier paquet avec une adresse IP destination correspondant à l'un des préfixes de la configuration de l'étaleur.

L'étaleur calcule alors les nouvelles adresses IP source et destination à l'aide d'une fonction cryptographique. Nous avons choisi le chiffrement AES, mais d'autres fonctions permettant le chiffrement de blocs de 128 bits pourraient être utilisées.

La fonction AES prend en entrée un bloc composé des identifiants d'interfaces des terminaux A et B, et la clef actuelle $K(t)$ dérivée du secret partagé. Le chiffrement fournit un bloc de 128 bits en sortie, que l'étaleur divise en deux blocs de 64 bits, utilisés pour réécrire les derniers 64 bits de IP_A et IP_B :

$$IID^1_{src} = AES(IID_A \vert IID_B, K(t))[0-63]$$
$$IID^1_{dst} = AES(IID_A \vert IID_B, K(t))[64-127]$$

Après la réécriture, la procédure habituelle de routage et de filtrage est appliquée au paquet. L'étaleur sauvegarde dans un contexte les adresses originales et les adresses réécrites. Cela termine l'étape *1a*.

Les adresses stables sont recalculées par la même méthode (AES est une fonction symétrique) par le second étaleur lors de l'étape *1b*. Après la réécriture, l'étaleur vérifie la validité de la somme de contrôle du protocole de transport.Si la somme est valide, le paquet est transmis pour suivre la politique habituelle de routage et de filtrage.

Si la somme de contrôle n'est pas valide, la cause peut être un problème de transmission. Une autre possibilité est une tentative d'un attaquant d'injecter un paquet sur le réseau, avec une adresse source forgée. En effet, les adresses IPv6 sont comprises dans le calcul de la somme de contrôle. L'attaquant ne pouvant pas connaître la réécriture effectuée par la fonction AES, son paquet aura probablement une somme de contrôle invalide après la réécriture.

Il est à noter que les paquets légitimes auront donc une somme de contrôle invalide lors du transit entre les deux étaleurs. Ce n'est pas un problème, car les nœuds intermédiaires n'ont pas à tester cette valeur. Cette somme sera valide lors de la vérification par les terminaux.

4.1.2 Analyse de la sécurité de l'étalement

Si un attaquant a connaissance de l'étalement, il peut tenter de deviner la différence apportée par le chiffrement AES sur la somme de contrôle. La somme étant codée sur 16 bits, il a une chance sur 65 536 de trouver la bonne valeur et donc d'injecter un paquet sur le réseau protégé par l'étaleur.

Ce paquet injecté aura cependant une adresse destination arbitraire du fait de la réécriture AES. Si on considère cette réécriture totalement aléatoire, les chances de cibler un terminal particulier sont de une sur 2^{64}, soit 2^{80} en ajoutant la protection de la somme de contrôle. Enfin, si l'attaquant souhaite forger un paquet avec une adresse source particulière, cela augmente encore la difficulté de 2^{64} pour un total d'une chance sur 2^{144}.

Il est de plus à noter qu'une valeur ne sera valide que pour une très courte durée. La prochaine valeur de $K(t)$ à $t+1$ changera le calcul AES et son implication sur la somme de contrôle.

4.2 Initialisation d'une connexion - réponse du réseau distant

L'objectif de la réécriture de l'adresse sur le premier paquet est d'invalider toute initialisation d'une connexion effectuée par un attaquant. Pour les paquets suivants, nous créons une séquence pour chaque flux de données, généré par une fonction g. Cette fonction g est un générateur aléatoire de séquences temporelles, un exemple pourrait être la fonction SHA1. g prend pour entrées l'heure actuelle, le secret partagé entre les réseaux, et une valeur longue de 64 bits (un identifiant d'interface).

Cette création débute à l'étape *2a*. Le second étaleur réécrit le premier paquet de réponse d'un terminal avec une adresse IP source aléatoire, et avec une valeur dérivée de l'adresse source du premier paquet pour la destination :

$$IID^2_{src} = random() \qquad IID^2_{dst} = g(t, secret, IID^1_{src})$$

Cette réécriture introduit une valeur aléatoire dans la séquence, mais le flux est toujours identifiable. En effet, l'adresse IP destination, réécrite avec des informations connues des deux étaleurs, permet la reconnaissance du flux.

À l'étape *2b*, le premier étaleur reconnaît l'adresse IP destination IP^2_{dst} avec l'aide du contexte enregistré à l'étape *1a*. Ce nouveau paquet est un acquittement de l'initialisation, l'étalement à l'aide de g peut alors commencer. L'étaleur sauvegarde la valeur de l'adresse IP source (rendu aléatoire en étape *2a*) et réécrit les adresses IP avec les valeurs statiques sauvegardées dans le contexte.

Cela termine l'étape 2. Le premier étaleur est désormais certain de l'initialisation de la connexion, et il peut utiliser la valeur aléatoire reçue pour initialiser une nouvelle séquence aléatoire.

4.3 Initialisation d'une connexion - Acquittement au second étaleur

L'étape *3a* débute au prochain paquet envoyé par le terminal émetteur du premier réseau. Les deux adresses sources et destination sont désormais étalées de la façon suivante :

$$IID^3_{src} = g(t, secret, IID^1_{src}) \qquad IID^3_{dst} = g(t, secret, IID^2_{src})$$

Au cours de l'étape *3b*, le second étaleur reconnaît le couple d'adresse étalé grâce à son contexte sauvegardé. Ce paquet est également l'acquittement de la valeur aléatoire

Étapes	IID Source réécrit	IID Destination réécrit
1a → *1b* SYN	$IID^1_{src} = AES(IID_A \vert IID_B, K(t))[0-63]$	$IID^1_{dst} = AES(IID_A \vert IID_B, K(t))[64-127]$
2a → *2b* SYN, ACK	$IID^2_{src} = random()$	$IID^2_{dst} = g(t, secret, IID^1_{src})$
3a → *3b* ACK	$IID^3_{src} = g(t, secret, IID^1_{src})$	$IID^3_{dst} = g(t, secret, IID^2_{src})$

Table 2: Réécriture des identifiants d'interface durant l'initialisation

envoyée en étape *2a*, cela confirme au second étaleur le succès de l'initialisation, qui est désormais terminée. Les étapes sont résumées dans le tableau 2.

4.4 Réécriture durant la durée de vie de la connexion

Après l'étape *3b*, les deux étaleurs suivent les séquences d'adresses $g(t, secret, IID^1_{src})$ et $g(t, secret, IID^2_{src})$. Cette réécriture est symétrique et les deux équipements terminaux reçoivent des adresses stables, leur comportement n'est pas impacté. Un attaquant est incapable d'injecter du trafic car il ne peut pas deviner la prochaine adresse utilisée.

5 Détails du traitement des paquets par les étaleurs

Notre description du protocole en section 4 est la situation idéale. Nous faisons l'hypothèse qu'aucune perte de paquets ne se produit, et la connexion utilise TCP. Cela aide à comprendre le fonctionnement, puisque notre protocole suit le même mécanisme de poignée de mains TCP.

Mais le protocole doit résister à la perte de paquets et à la retransmission de données. De même, nous devons être compatibles avec les autres protocoles de la couche transport qui ne suivent pas une procédure rigoureuse d'initialisation, comme UDP ou ICMP. Nous détaillons donc le traitement des paquets par les étaleurs, qui résiste à la perte de paquets et est indépendant du protocole.

5.1 Étapes détaillées du traitement des paquets sortant vers l'Internet

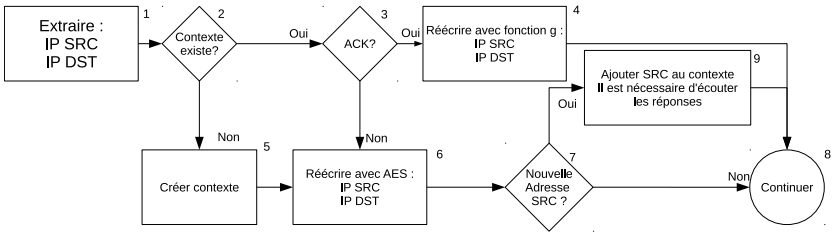

Figure 4: Traitement d'un paquet sortant vers l'Internet

Le traitement des paquets du réseau local vers l'Internet est décrit par la figure 4. Pour chaque paquet sortant, l'étaleur extrait le couple (IP_{src}, IP_{dst}) (étape 1). Il vérifie

si un contexte existe (étape 2), et si un acquittement est déjà reçu (étape 3). Si les deux conditions sont réunies, nous sommes dans le cas d'une connexion établie et il est possible de réécrire les adresses à l'aide de la fonction g.

Si le contexte n'existe pas encore, il est nécessaire d'en créer un. Ce contexte contient les adresses sources et destinations statiques (étape 5). L'étaleur continue en étape 6 par la réécriture des adresses IP_{src} et IP_{dst} à l'aide du chiffrement AES et de la clef $K(t)$.

Comme l'adresse source réécrite est un paramètre pour la réponse du second réseau, il est nécessaire de la sauvegarder (étape 7 et 9). Il est parfois nécessaire de conserver plus d'une adresse si plusieurs paquets sont émis avant la réception d'un acquittement.

Dans tous les cas, le paquet modifié est ensuite soumis aux règles habituelles.

5.2 Étapes détaillées du traitement des paquets provenant d'Internet

La figure 5 décrit le traitement pour tous les paquets provenant de l'Internet. Cela commence par l'extraction du couple (IP_{src}, IP_{dst}). L'objectif est de savoir s'il existe un contexte pour ce couple (étape 2 et 3). Si un contexte est trouvé, il peut-être nécessaire de mettre à jour le drapeau signalant l'acquittement (étape 8). Les adresses IP d'un paquet avec contexte sont ensuite réécrites avec les valeurs statiques (étape 5), et le paquet est transmis aux règles de routage et de filtrage habituelles (étape 6).

Si aucun contexte n'est enregistré, il est nécessaire de déchiffrer les adresses avec la fonction AES et la clef $K(t)$ (étape 7). Ce déchiffrement est suivi par la vérification de la somme de contrôle de la couche transport (étape 9). Si la somme est invalide, le paquet est détecté comme forgé et rejeté. Dans le cas contraire, l'étaleur initialise un contexte (étape 10) et le paquet est transmis aux règles de routage et de filtrage habituelles.

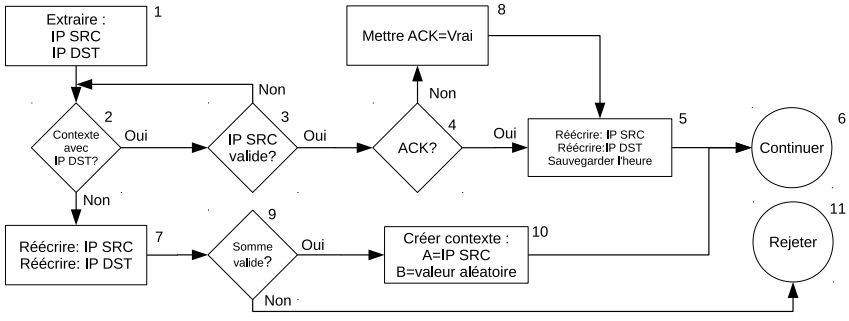

Figure 5: Traitement d'un paquet reçu depuis Internet

6 Travaux reliés

Il existe des travaux antérieurs sur les adresses IPv6 dynamiques. Le plus proche est "Moving target IPv6 Defense" [DGU+11]. Cette solution utilise des tunnels UDP pour varier fréquemment les adresses, les vrais paquets sont encapsulés. Un chiffrement est optionnel pour protéger le contenu du paquet. La principale différence avec nos travaux est le choix d'un tunnel pour encapsuler les paquets, ce qui engendre des problèmes de taille maximale et a des conséquences sur la bande passante.

Une autre approche est "An Architecture for Network Layer Privacy" [BGMA07]. Les auteurs utilisent une extension de SHIM6 pour changer d'adresse fréquemment. Le terminal doit allouer toutes les adresses nécessaires à la communication avant de l'utiliser, ce qui consommera beaucoup de ressources sur le terminal. Cela implique également d'envoyer de très nombreux paquets de découverte de voisins avant le début de la connexion.

Toujours avec SHIM6, un papier [CBL] protège les services critiques d'un déni de service. L'idée est de changer l'adresse d'une connexion attaquée vers une autre adresse non attaquée, pour toutes les connexions déjà établies.

7 Conclusion et travaux futurs

L'étalement d'adresse est une solution innovante pour identifier une connexion. C'est un nouveau mécanisme pour protéger des usurpations, à la fois lors de l'initialisation d'une connexion et durant sa durée de vie. Nous proposons un protocole capable d'initialiser une séquence d'adresses pour chaque flux, ainsi que le processus à suivre sur les étaleurs.

Nous avons développés une implémentation de référence, notamment pour évaluer la durée de vie d'une adresse. Par manque de place, nous ne pouvons pas détailler. Nos analyses préliminaires montrent cependant qu'il est réaliste de changer d'adresse toutes les 200 millisecondes, avec une fenêtre temporelle de 100 ms durant laquelle deux adresses sont acceptées (pour prendre en compte la latence du réseau et les désynchronisations). Cette configuration n'a en effet aucun impact sur les réseaux réels que nous avons testés.

Nos travaux futurs concernent l'évaluation de la solution, notamment les performances nécessaires à la détection de paquets forgés. Ces paquets ne sont détectés qu'après un déchiffrement AES et un calcul de somme de contrôle, ce qui est à comparer avec des solutions comme IPsec. Une évaluation des différents algorithmes pour générer les séquences d'adresses est également à considérer.

Références

[Aur05] T. Aura. Cryptographically Generated Addresses (CGA). RFC 3972 (Proposed Standard), March 2005. Updated by RFCs 4581, 4982.

[BGMA07] M. Bagnulo, A. Garcia-Martinez, and A. Azcorra. An architecture for network layer privacy. In *Communications, 2007. ICC '07. IEEE International Conference on*, pages 1509–1514, June 2007.

[BMR99] N. Brownlee, C. Mills, and G. Ruth. Traffic Flow Measurement : Architecture. RFC 2722 (Informational), October 1999.

[CBL] Xiangbin Cheng, Jun Bi, and Xing Li. Swing - a novel mechanism inspired by shim6 address-switch conception to limit the effectiveness of dos attacks. In *Networking. ICN 2008. Seventh International Conference on*, pages 267–272.

[DGU+11] Matthew Dunlop, Stephen Groat, William Urbanski, Randy Marchany, and Joseph Tront. Mt6d : A moving target ipv6 defense. In *MILITARY COMMUNICATIONS CONFERENCE - MILCOM*, pages 1321–1326, Nov 2011.

[FS00] P. Ferguson and D. Senie. Network Ingress Filtering : Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827 (Best Current Practice), May 2000. Updated by RFC 3704.