SEND : la découverte de voisins IPv6 sécurisée

Francis Dupont
Internet Systems Consortium
Francis.Dupont@fdupont.fr

7 novembre 2007

Résumé

SEND est la version sécurisée du protocole de découverte des voisins d'IPv6 qui gère entre autres la correspondance entre les adresses niveaux réseau et liaison, et l'auto-configuration.

Il permet en particulier de contrer les usurpations d'adresses et l'injection de préfixes ou de routeurs pirates.

Il utilise la cryptographie d'une manière non immédiate.

Abstract

SEND is the secure version of the IPv6 neighbor discovery protocol which manages among other things the mapping between network layer and link layer addresses, and the auto-configuration.

In particular it provides a defense against address, prefix or router spoofing.

Its usage of the cryptography is quite sophisticated.

Très courte introduction à IPv6

IPv6 [1] est la nouvelle version du protocole réseau de l'Internet. Sa principale caractéristique, mais pas la seule, est d'offrir un adressage sur 128 bits au lieu des 32 bits d'IPv4.

Les adresses IPv6 ont une structure : les adresses unicast commencent par 64 bits de préfixe utilisés par le routage, suivis par 64 bits d'identifiant d'interface (IID) qui est par défaut l'adresse IEEE MAC sur 64 bits avec le bit U/L inversé. Elles ont aussi une portée, par exemple les adresses `fe80::<iid>` sont les adresses locales au lien (*link-local*).

Pour finir la réponse à la question que tout le monde pose est oui, IPv6 va être déployé :

- il ne reste que pour quelques années d'adresses IPv4 à allouer ;
- le logo "Vista Ready" exige le support d'IPv6, quoique Teredo soit présenté par Microsoft plutôt comme un "peer-to-peer"...
- quelques très gros utilisateurs doivent disposer d'un espace d'adresses privées en dizaines de millions d'adresses, donc trop gros pour les plages d'adressage réservées (10.0.0.0/8, 192.168.0.0/16, etc) ;
- tout ceux qui savent vraiment ce qu'est un NAT (*Network Address Translator*) n'en veulent plus, surtout pour eux !

La découverte des voisins

Le protocole de découverte des voisins d'IPv6 [2] fait bien plus que remplacer le protocole ARP [3] dont en passant la sécurisation est totalement impossible.

En effet les fonctions assurées par le protocole de découverte des voisins (ND) sont :

- la découverte des routeurs par défaut sur le lien ;
- la découverte des préfixes du lien ;
- la découverte des paramètres du lien (*timers, hop limit, MTU, ...*) ;
- l'auto-configuration des adresses y compris la vérification qu'une adresse n'est pas déjà utilisée (*Duplicate Address Detection*, DAD) [4] ;
- la résolution d'adresse (la détermination de l'adresse lien en fonction de l'adresse IPv6) ;
- le routage sur le lien (la détermination du *next-hop*) ;
- la détection des voisins inaccessibles (*Neighbor Unreachability Detection*, NUD) ;
- la redirection (sur le lien, vers un autre routeur, ...).

Le protocole utilise cinq types de messages :
- la sollicitation du routeur (RS) ;
- l'annonce du routeur (RA) ;
- la sollicitation du voisin (NS) ;
- l'annonce du voisin (NA) ;
- la redirection (RED).

Des options extensibles transmettent les adresses niveau liaison, les informations sur un préfixe, etc.

L'analyse de la sécurité de ND [5] démontre que la plupart des fonctions de ND sont facilement attaquables. L'objectif de SEND est de protéger les points les plus critiques, en particulier les adresses et la découverte des préfixes.

Exemples de découverte des voisins

Pour les trois premières fonctions (découverte des routeurs, des préfixes et des paramètres), ND utilise un échange RS/RA :

```
sendr# tcpdump -e -p -v -n -s 1500 -i lnc1
tcpdump: listening on lnc1, link-type EN10MB (Ethernet), capture size 1500 bytes
14:07:43.208230 00:0c:29:cd:45:03 > 33:33:00:00:00:02, ethertype IPv6:
 fe80::20c:29ff:fecd:4503 > ff02::2: icmp6: router solicitation
 (src lladdr: 00:0c:29:cd:45:03)
14:07:43.264477 00:0c:29:e8:f8:76 > 33:33:00:00:00:01, ethertype IPv6:
 fe80::20c:29ff:fee8:f876 > ff02::1: icmp6: router advertisement
 (chlim=64, pref=medium, router_ltime=1800, reachable_time=0, retrans_time=0)
 (src lladdr: 00:0c:29:e8:f8:76)
 (prefix info: LA valid_ltime=2592000,preferred_ltime=604800,prefix=2001:fd::/64)
```

Mais l'échange le plus fréquent est le NS/NA pour le NUD (vérification de la présence des voisins) :

```
14:17:28.944505 00:0c:29:cd:45:03 > 00:0c:29:e8:f8:76, ethertype IPv6:
 2001:fd::20c:29ff:fecd:4503 > 2001:fd::20c:29ff:fee8:f876: icmp6: neighbor sol:
 who has 2001:fd::20c:29ff:fee8:f876
 (src lladdr: 00:0c:29:cd:45:03)
14:17:28.944654 00:0c:29:e8:f8:76 > 00:0c:29:cd:45:03, ethertype IPv6:
 2001:fd::20c:29ff:fee8:f876 > 2001:fd::20c:29ff:fecd:4503: icmp6: neighbor adv:
 tgt is 2001:fd::20c:29ff:fee8:f876 (RS)
```

Le problème des adresses

Un des principaux problèmes est le vol d'adresse dans toutes ses variantes. Pour le contrer il faut lier une adresse, ou ses 64 derniers bits après le préfixe

(l'identifiant d'interface, *IID*), à une clé afin de signer les messages critiques.

Deux grandes méthodes sont possibles :

- dériver la clé de l'adresse (*Address Based Key*) en utilisant un protocole cryptographique de type IBE (*Identity Based Encryption*, [6]). Un mécanisme basé sur le *pairing* de Weil dans des courbes elliptiques a été proposé mais n'a pas été retenu, peut-être parce qu'il n'était pas "assez mûr" ? Un autre problème de ce type de solutions est qu'il nécessite un tiers de confiance, l'*Identity-based Private Key Generator*, alors que les concepteurs de la protection des adresses dans SEND recherchaient un mécanisme strictement sans état.
- dériver l'adresse de la clé (*Key Based Address*). C'est la solution qui a été retenue sous le nom de CGA (*Cryptographically Generated Address*, [7]) et qui est décrite dans la section suivante.

Les CGA

Dans une CGA, l'IID est dérivée d'une clé publique RSA et d'un jeu de paramètres. La clé privée associée sert à signer les messages. La validation des messages est effectuée d'abord sur les règles de dérivation et ensuite sur la signature qui est une opération coûteuse.

Du point de vue cryptographique le problème est de fournir une preuve de possession de l'adresse tout en ayant par défaut pas d'état préalable dans les récepteurs.

Les paramètres sont :

- un modificateur sur 128 bits ;
- le préfixe sur 64 bits ;
- un compteur de collision sur 8 bits prenant les valeurs 0, 1 ou 2 ;
- la clé publique (encodage DER de la structure ASN.1 d'X.509) ;
- d'éventuelles extensions.

Ils servent à calculer deux valeurs de hachage :

- $hash_1$: les 64 premiers bits de l'application de SHA-1 aux paramètres ;
- $hash_2$: les 112 premiers bits avec le préfixe et le compteur de collision mis à zéro.

$hash_1$ fournit l'IID à l'exception des 3 premiers bits qui forment le paramètre de sécurité *sec*, du bit U (*universal*) forcé à un et du bit G (*group*) forcé à zéro.

La création d'une CGA est simple :

1. tirage d'un couple clé publique/clé privée ;
2. tirage d'une valeur aléatoire pour le modificateur ;
3. calcul de $hash_2$ et incrémentation du modificateur tant que $hash_2$ ne convient pas ;
4. remplissage du jeu de paramètres avec le compteur de collision à 0 ;
5. calcul de $hash_1$ et dérivation de l'adresse ;

6. vérification de l'unicité de l'adresse sur le lien, sinon incrémentation du compteur de collision et reprise à la phase précédente.

$hash_2$, qui ne dépend pas du préfixe afin de faciliter la renumérotation, doit avoir ses $16sec$ premiers bits à zéro. L'idée est d'augmenter le coût d'une attaque contre $hash_1$, c'est-à-dire trouver un jeu de paramètres donnant le même IID, du même facteur : la complexité du problème est en $O(2^{59+16sec})$ si SHA-1 reste robuste pour les problèmes de pré-images. Trouver une bonne valeur du modificateur, c'est-à-dire créer une nouvelle CGA, est aussi en $O(2^{16sec})$ ce qui est délibérément très difficile pour les grandes valeurs de *sec*.

Ce mécanisme de *puzzle* a été rendu nécessaire par le relatif petit nombre de bits disponible dans l'IID, la preuve de possession d'une adresse étant basée sur la difficulté à "casser" le hachage ou la clé privée.

L'utilisation de clé nue est due à la contrainte sans état mais n'est pas incompatible avec un système à base de certificats, il faut seulement qu'il ne soit pas "par défaut".

SEND

La version sécurisée de ND, SEND (*SEcure Neighbor Discovery*, [8] et [9]), est basée sur plusieurs mécanismes : des estampilles temporelles, des nonces, des preuves de possession des adresses et des délégations autorisées de préfixe avec des ancres de confiance.

Sauf pour le dernier mécanisme, de nouvelles options ont été définies :
- estampille temporelle (*timestamp*)
- nonce
- jeu de paramètres CGA
- signature RSA (avec le hachage de la clé publique et portant sur les adresses, les en-têtes ICMPv6 et ND, et toutes les options jusqu'à l'option signature/dernière option non comprise)

Les estampilles temporelles

Les estampilles permettent de rejeter les messages trop dans le passé ou le futur, ou pour une source connue, c'est-à-dire déjà présente dans le cache des voisins, trop décalés par rapport à la différence des horloges relevée au message précédent (les paramètres standards sont un delta maximal de 5 minutes, une imprécision d'horloge d'une seconde et un décalage maximal de 1 pour cent).

Plus précisément :

$$-Delta < (RDnew - TSnew) < +Delta$$

$$TSnew + fuzz > TSlast + (RDnew - RDlast) \times (1 - drift) - fuzz$$

Les nonces

Les échanges sollicitation/annonce sont protégés par des nonces ce qui assure un anti-rejeu : les réponses qui ne correspondent pas à une requête en cours sont rejetées.

Comme pour les estampilles temporelles, l'état nécessaire pour pouvoir valider les messages n'est qu'ajouté aux entrées du cache des voisins, c'est-à-dire il n'est pas créé et géré spécifiquement.

Les adresses

Les nœuds supportant SEND utilisent des CGA et transmettent le jeu de paramètres dans une option. Les routeurs utilisent en outre des certificats plus riches que des clés nues.

Les signatures

Tous les messages, à l'exception des RS de source indéfinie, doivent être signés avec une clé privée associée à l'adresse (CGA et/ou certificat). Il est donc impossible d'usurper les adresses car ce mécanisme fournit une preuve de possession (*ownership*) de l'adresse.

Les certificats

Les routeurs utilisent des certificats de clé publiques X.509v3 avec des attributs décrivant les préfixes gérés [10]. Ces attributs ont été définis pour le support de S-BGP, une version sécurisée du protocole de routage inter-domaine de l'Internet.

Par exemple :

```
sendr# openssl x509 -in cert.pem -text
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 0 (0x0)
        .......
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            .......
            sbgp-ipAddrBlock: critical
                IPv6 (Unicast):
                  2001:fd:/64
            .......
```

La chaîne de délégation des préfixes et la chaîne de signature des certificats sont parallèles : une délégation est donc reconnue comme sûre dès qu'une ancre de confiance figure dans le chemin de certification.

La découverte des délégations autorisées

SEND comprend un protocole sollicitation/annonce pour transmettre le chemin de certification en fonction d'ancres de confiance (désignées par leur sujets X.501 `Subject` ou FQDN `SubjectAltName`).

Il suffit donc de configurer sur les machines une ancre de confiance compatible pour sécuriser dans l'auto-configuration la partie découverte des préfixes du lien.

Analyse critique de SEND

SEND ne protège pas des attaques contre une couche liaison non sécurisée, par exemple il n'assure pas que les paquets proviennent du même nœud que les messages ND, ni lie cryptographiquement les adresses liaison et réseau.

De même le service multicast est géré par MLD (*Multicast Listener Discovery*, [11]) qui n'est pas (encore) sécurisé, alors que ND et SEND reposent fortement sur ce service.

SEND protège contre la création de fausses entrées dans le cache des voisins, la non-détection des voisins inaccessibles, les dénis de service utilisant la détection de la duplication des adresses, les fausses annonces provenant d'un routeur pirate (mais pas totalement d'un routeur piraté) et les rejeux.

De plus, SEND a été conçu pour être raisonnablement robuste.

SEND a quand même quelques défauts de conception notables :

- la cryptographie utilisée n'est pas "agile" : RSA et surtout SHA-1 ne sont pas remplaçables sans redéfinir le protocole ;
- le mélange de SEND avec un ND standard et donc non sécurisé sur le même lien est défini mais fait perdre le plus gros des avantages de SEND ;
- l'utilisation intensive des signatures rend SEND sensible aux dénis de service quand certains nœuds sont limités en performances de calcul ;
- le mécanisme de délégation autorisée des préfixes est complexe à déployer. Sans lui, il suffit de pré-configurer les nœuds quite à perdre toute flexibilité.

Ces défauts restent relatifs : SHA-1 est faible sur les collisions, pas sur les pré-images ; le coût des signatures n'est que le coût de la sécurité ; l'APNIC (le RIR Asie-Pacifique) expérimente les attributs X.509 du RFC 3779 [10].

SEND semble incontournable pour maintenir un niveau de sécurité acceptable dans de nouveaux protocoles comme NETLMM (*Network-based Localized Mobility Management*, [12]) ; et pour finir SEND est au minimum recommandé pour les infrastructures avec de hautes exigences de sécurité.

Retour d'expérience sur des implémentations de SEND

Nous disposons de deux implémentations expérimentales de SEND, une écrite à l'ENST Bretage fin 2005 mais non diffusée, l'autre par le laboratoire de DoCoMo aux USA.

Les deux sont entièrement en mode utilisateur : les paquets sont détournés dans les couches basses pour être traités à l'extérieur du noyau. Non seulement l'impact sur les performances est important, mais certains détails ne peuvent pas être correctement supportés, par exemple une requête avec une estampille temporelle inacceptable est simplement ignorée alors que dans certains cas il faudrait y répondre. En général, la gestion du mélange de SEND avec le ND standard est très partielle car elle demande une action directe sur le cache des voisins maintenu par et dans le noyau.

Enfin la partie certificat reste complexe à gérer quoique la dernière version d'OpenSSL (0.9.8e) supporte le RFC3779 (quand elle compilée avec la configuration idoine).

Conclusion

Le protocole de découverte des voisins est par conception protégé de l'activité en dehors du lien : il utilise des adresses dont la portée est le lien et vérifie que les messages n'ont pas été routés.

Avec un peu de configuration sur toutes les machines, il est facile de filtrer les mauvaises annonces venant de nœuds se prenant par "stupidité" pour des routeurs. Malheureusement ce type de problèmes est surtout fréquent dans les réseaux peu ou pas gérés, ce qui contredit l'aspect configuration.

En prenant en compte la contrainte d'éviter au maximum d'entretenir de l'état, en particulier sur les routeurs, afin de gérer la sécurité, la sécurité offerte par SEND semble raisonnable. Une solution plus directe comme une association de sécurité IPsec par couple de voisins serait probablement plus sûre et sûrement bien plus coûteuse, sans parler de la difficulté de baser l'identification des nœuds sur des adresses dont la gestion, y compris l'auto-configuration, est un des rôles du protocole...

Pour finir SEND a clairement besoin de plus d'expérimentation et de déploiement. En effet les contraintes de conception et les compromis qui en résultent n'ont pas vraiment été validés dans le monde réel, du moins dans le monde réel civil.

Références

[1] S. Deering, R. Hinden, *Internet Protocol Version 6 (IPv6) Specification*, RFC 2460, IETF, Décembre 1998

[2] T. Narten, E. Nordmark, W. Simpson, *Neighbor Discovery for IP Version 6 (IPv6)*, RFC 2461, IETF, Décembre 1998

[3] D. Plummer, *An Ethernet Address Resolution Protocol*, RFC 826, IETF, Novembre 1982

[4] S. Thomson, T. Narten, *IPv6 Stateless Address Autoconfiguration*, RFC 2462, IETF, Décembre 1998

[5] P. Nikander (ed.), J. Kempf, E. Nordmark, *IPv6 Neighbor Discovery (ND) Trust Models and Threats*, RFC 3756, IETF, Mai 2004

[6] A. Shamir, *Identity-Based Cryptosystems and Signature Schemes*, Crypto 84, Springer-Verlag LNCS 196, 1984

[7] T. Aura, *Cryptographically Generated Addresses (CGA)*, RFC 3972, IETF, Mars 2005

[8] J. Arkko (ed.), J. Kempf, B. Zill, P. Nikander, *SEcure Neighbor Discovery (SEND)*, RFC 3971, IETF, Mars 2005

[9] J. Arkko, T. Aura, J. Kempf, V.-M. Mäntylä, P. Nikander, M. Roe, *Securing IPv6 Neighbor and Router Discovery*, 1st ACM Workshop on Wireless Security, Atlanta, Septembre 2002

[10] C. Lynn, S. Kent, K. Seo, *X.509 Extensions for IP Addresses and AS Identifiers*, RFC 3779, IETF, Juin 2004

[11] S. Deering, W. Fenner, B. Haberman. *Multicast Listener Discovery (MLD) for IPv6*, RFC 2710, IETF, Octobre 1999

[12] V. Narayanan, J. Soininen,
`http://www.ietf.org/html.charters/netlmm-charter.html`