

Centre
Interuniversitaire de
Ressources
Informatiques de
Lorraine

IPv6-ADIRE

Mise en oeuvre d'un service DNS avec Dynamic Updates (Dynamic DNS)

- Octobre 2005 -

Introduction

Etant donné la longueur des adresses IPv6, l'administration du service DNS peut très vite devenir une tâche fastidieuse notamment pour ce qui concerne les enregistrements inverses. L'utilisation d'un service DNS avec "Dynamic Updates" [1] aussi appelé DDNS (Dynamic DNS) permet à un client de s'enregistrer automatiquement par l'envoi de messages spécifiques au serveur. Ce document qui s'inscrit dans le cadre du projet **IPv6-ADIRE**[1] va donc présenter le travail nécessaire pour la mise en oeuvre de tels mécanismes du côté serveur et client sans négliger l'aspect sécurité [2].

1 – Configuration du serveur de noms

Les extraits de configurations qui suivent sont basés sur le serveur Bind dans sa version 9.3.1. Les versions antérieures (ex : Bind 8) supportent également les mises à jour dynamiques mais les configurations peuvent s'avérer différentes; c'est pourquoi il sera préférable de se reporter aux documentations adaptées [3].

La configuration du serveur ne nécessite que très peu de changements pour activer cette fonctionnalité intéressante. Il suffit pour cela de définir sa politique de mises à jour dynamiques pour la zone concernée dans le fichier named.conf :

```
zone "postes.ipv6.ciril.fr" IN {
  type master;
  file "postes.ipv6.ciril.fr.db";
  // On autorise les maj automatiques pour cette zone
  update-policy {
    // On autorise la modification des enregistrements AAAA
    grant <key> subdomain postes.ipv6.ciril.fr AAAA;
  };
};
```

La politique de mise à jour définie ci-dessus permet à un poste possédant la clé <key> de modifier les enregistrements de type AAAA pour le sous-domaine postes.ipv6.ciril.fr. La section update policy peut définir ainsi plusieurs règles du type :

```
grant <key> <type> <zone> <record-types>;
```

La balise <key> fait référence à une clé explicitement définie dans le fichier named.conf (cf. Génération des clés). Il est également possible de définir les clés dans un fichier séparé (exemple : /var/named/keys.conf) puis d'inclure ce dernier dans le fichier de configuration du démon :

```
include "keys.conf";
```

La balise <type> représente la portée de la règle. En d'autres termes la règle peut s'appliquer à un sous-domaine (subdomain) dans son ensemble ou uniquement à un enregistrement simple (name).

```
grant host1.ipv6.ciril.fr.key. name host1.ipv6.ciril.fr. AAAA;
grant host1.ipv6.ciril.fr.key. name host1.ipv6.ciril.fr. AAAA;
grant ipv6.ciril.fr.key.  subdomain ipv6.ciril.fr. AAAA;
```

[1] **IPv6-ADIRE** : Projet lancé par le ministère de l'Education Nationale, de la Recherche et de l'Enseignement Supérieur visant à déployer IPv6 sur des réseaux de production.

Finalement la balise <record-types> permet de définir les types d'enregistrements modifiables. Il ne serait par exemple pas très judicieux de permettre la modification d'enregistrements MX. De la même façon il sera préférable d'éviter d'activer les mises à jour dynamiques dans une zone contenant des enregistrements critiques (serveurs, …).

Génération des clés :

Afin de garantir l'authentification des différents acteurs, il est donc nécessaire de générer les clés qui permettront de signer les messages de mises à jour des enregistrements. La signature des transactions (TSIG : Trasaction SIGnature [4]) permettra ainsi d'éviter que n'importe quel poste puisse effectuer des modifications dans les enregistrements du serveur de noms.

La génération des clés publiques/privées est réalisée par l'intermédiaire de l'outil dnssec-keygen fournit avec BIND :

```
# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST updates.postes.
```

Suite à l'exécution de cette commande deux fichiers contenant respectivement la clé publique et la clé privée sont créés. (Attention : positionner les droits de lecture) Il est donc maintenant nécessaire de déclarer cette clé privée dans le fichier de configuration named.conf (ou keys.conf). C'est le partage de cette clé entre le serveur et le poste client qui va permettre d'authentifier les deux acteurs lors des transactions.

```
key ipv6.ciril.fr.key. {
      algorithm "hmac-md5";
      secret "wMkRFcweTFSBwlbWDaoF0w==";
};
```

La clé secrète définie dans la section key est celle qui a été généré dans le fichier Kupdate.postes.+157+56324.private.

```
# cat Kupdate.postes.+157+56324.private
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: wMkRFcweTFSBwlbWDaoF0w==
```

2 – Configuration d'un poste client

Il est maintenant nécessaire de copier le fichier contenant la clé privée (.private) générée sur le serveur par le biais de la commande dnssec-keygen sur le poste client. Cette copie devra être sécurisée afin d'éviter qu'une personne mal intentionnée puisse récupérer cette clé et ainsi réaliser des mises à jours dynamiques non souhaitées sur le serveur. Pour les mêmes raisons, il faudra veiller à convenablement positionner les droits sur ce fichier pour que ce dernier ne soit pas lisible par n'importe quel utilisateur.

La commande qui permet de réaliser les "updates" dynamiques sur un poste Linux est nsupdate. Son fonctionnement (man nsupdate) est très simple :

```
# nsupdate -k ./Kkey-dns.+157+20468.private
Creating key...
>  update delete host1.postes.ipv6.ciril.fr
> update add host1.postes.ipv6.ciril.fr 60  IN AAAA <IPv6_ADDR>
> send
```

Cette séquence permet d'effacer l'enregistrement host1.postes.ipv6.ciril.fr sur le serveur concerné puis d'ajouter un enregistrement IPv6 pour cette même ressource. Il est bien évidemment possible de réaliser de la même façon l'enregistrement inverse pour cette ressource. Pour des questions de sécurité il est indispensable que les horloges des postes clients et du serveur soient synchronisées (à 5 minutes près). Dans le cas contraire un message vous avertira que la commande a échoué.

L'utilisation d'un script permet de faciliter le déploiement d'une solution à base de DNS dynamique. Voici un exemple de script permettant d'effectuer l'enregistrement dynamique d'un poste IPv6 :

```
% cat update.sh
#!/bin/sh
HOSTNAME=`hostname`
DOMAIN=postes.ipv6.ciril.fr
IPADDR=`/sbin/ifconfig eth0|grep 'inet6'|grep 'Global'|tr -s " "|cut -d/ -
f1|cut -d" " -f4`
echo "server sargas.ciril.fr" > /etc/nsupdate
echo "prereq nxrrset $HOSTNAME.$DOMAIN IN AAAA" >> /etc/nsupdate
for i in $IPADDR
do
   echo "update add $HOSTNAME.$DOMAIN 60  IN AAAA $i" >> /etc/nsupdate
done
echo "" >> /etc/nsupdate
/usr/bin/nsupdate -k Kupdate.postes.+157+56324.private -d /etc/nsupdate
```

La directive *prereq nxrrset* permet de tester la non préexistence d'une ressource possédant le même nom. En d'autres termes, si l'enregistrement testé existe déjà alors l'ajout suivant ne sera pas réalisé.

Les enregistrements dynamiques n'apparaissent pas dans les fichiers de zones sur le serveur. En effet, ces derniers sont stockés dans des des fichiers .jnl aux côtés des fichiers de zones déjà existants. Le redémarrage du serveur n'entraîne pas la suppression de ces fichiers. Si vous désirez donc effacer tous les enregistrements dynamiques pour une zone donnée il vous suffira alors de supprimer le fichier .jnl correspondant. Pour supprimer uniquement un seul enregistrement vous devrez utiliser la commande nsupdate.

Remarque : À la rédaction de cette documentation, il semblerait que seuls les systèmes Linux et BSD disposent d'un outil (nsupdate) permettant de réaliser des enregistrements dynamiques pour des ressources IPv6 vers un serveur de noms IPv6 (transport des messages de mises à jour en IPv6).

Conclusion

L'inconvénient majeur de la mise en oeuvre d'un mécanisme de Dynamic DNS réside dans le fait qu'aucun système d'exploitation ne permet à l'heure actuelle de réaliser cette tâche directement du côté client. Cette extension du service DNS sensée faciliter la tâche d'administration s'avère donc assez contraignante pour le moment (diffusion des clés secrètes). Une alternative possible consisterait à coupler l'utilisation de ce mécanisme à l'utilisation d'un service DHCPv6 (draft-ietf-dhc-dhcpv6-fqdn [5]). A l'heure actuelle aucune implémentation de DHCPv6 ne permet d'exploiter cette fonctionnalité cependant Dibbler [6] prévoit d'intégrer ce mécanisme.

Références

RFC :

[1] RFC 2136 - Dynamic Updates in the Domain Name System (DNS UPDATE)

[2] RFC 2137 - Secure Domain Name System Dynamic Update

[4] RFC 2845 - Secret Key Transaction Authentication for DNS (TSIG)

Documents :

[5] draft : The DHCPv6 Client FQDN Option

Sites Internet :

[3] BIND (Berkeley Internet Name Domain)
http://www.isc.org/index.pl?/sw/bind/

[6] Dibbler - a portable DHCPv6
http://klub.com.pl/dhcpv6/