DNS

principe et mise en place

EBC Informatique

Centre de compétences UNIX

TABLE DES MATIERES

Notes préliminaires :

Tout en tentant de rester le plus général possible dans cette étude, je tiens à vous informer que la totalité des manipulations UNIX expliquées ont été effectuées sur un serveur fonctionnant sous distribution RedHAT 7.0.

Deux serveurs DHCP ont été testés. Le premier, livré avec la distribution RedHAT 7.0, est suffisant si l'on ne désire pas la prise en charge du DNS dynamique. Le second, téléchargeable sur le site de l'Internet Software Consortium, est obligatoire si l'on désire mettre en place le DNS dynamique. Sa version la plus récente au moment de l'écriture de ces lignes est DHCPd 3.0rc7. Les versions plus récentes ne devraient pas poser de problèmes particulier par rapport aux informations de cette étude.

Le serveur de noms (serveur DNS) utilisé est celui livré avec la distribution RedHAT 7.0. Sa version exacte est : BIND 8.2.2_P5. Les versions plus récentes, téléchargeables sur le site de l'Internet Software consortium http://isc.org/, ne devraient pas poser de problèmes particulier par rapport aux informations de cette étude.

Concernant les systèmes Windows, j'ai testé l'ensemble des configurations réseau décrites avec Microsoft Windows 98 et Microsoft Windows NT 4 (SP 6).

I] Qu'est ce que le DNS ?

a) Introduction

Un serveur de noms permet d'associer une adresse IP à un nom. Dans un réseau, chaque machine se voit attribuer une adresse IP unique, qui permet de l'identifier. C'est un peu comme une adresse postale, qui permet d'identifier une maison de façon certaine. Mais si une adresse chiffrée est plus facile à manipuler par un ordinateur, elle est difficile à mémoriser par un humain. Ainsi, on se souvient plus facilement de www.linux-france.org, mais plus difficilement de 216.167.114.128. Le serveur de noms va permettre de trouver l'adresse IP à partir d'un nom (ou inversement), que l'ordinateur pourra ensuite interroger.

Un serveur de noms permet aussi de faciliter la distribution des mails dans un réseau local, permet de mettre plus facilement en place des alias sur des noms de machines, et offre la mémorisation des adresses résolues par un autre serveur de noms (comme celui du fournisseur d'accès, par exemple).

Le terme DNS (Domain Name System) fait référence au système de résolution des noms.

b) Le DNS : la théorie

Avant d'entrer dans le vif du sujet, il peut être intéressant d'expliquer le principe de fonctionnement du DNS. Pour des raisons de commodité, il est plus facile pour un humain de manipuler des noms significatifs, tels que http://www.linux-france.org/, que des adresses codées sur plusieurs octets pour une machine (comme 216.167.114.128).

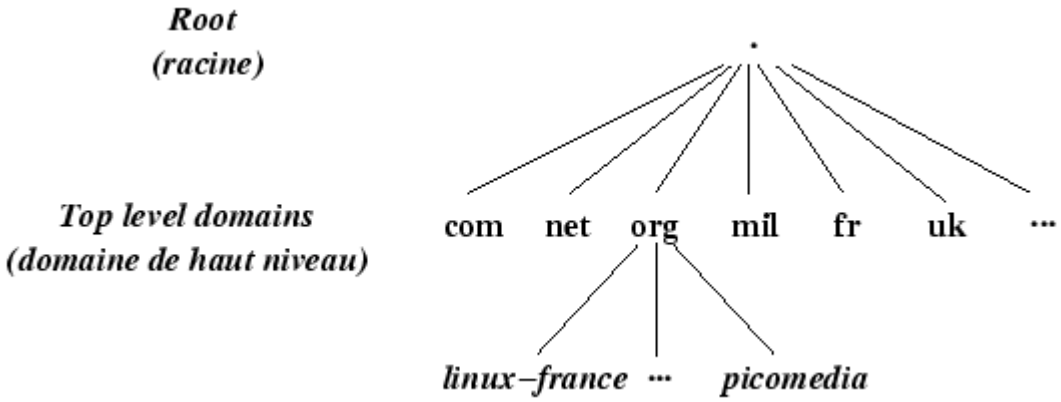
Au début des années 1970, un système centralisé à Stamford avec un fichier global HOST.TXT global a fonctionné. On pouvait récupérer la version la plus récente de ce fichier par FTP, ce qui n'était pas pratique. Avec l'explosion du nombre de machines connectées à l'Internet (on estime que plusieurs machines dans le monde apparaissent chaque minute), ce système est devenu totalement ingérable. Un modèle centralisé sur un serveur est donc impossible à mettre en place, du au nombre d'hôtes et le nombre de mises à jour nécessaires à apporter.

Le DNS (Domain Name System, Système de Nom de Domaine en français) a été conçu pour résoudre ce problème, en proposant un modèle hiérarchisé.

Chaque machine (terminal, serveur ...) reliée à un réseau se voit attribuer un petit nom. Ce nom est unique dans le domaine auquel elle appartient. Ainsi, pour les domaines domaine1.com et domaine2.com, on peut avoir deux machines portant des noms similaires ou différents. Par exemple, mail.domaine1.com et mail.domaine2.com désignent deux machines différentes, d'adresses IP différentes.

On peut comparer cette adresse à une adresse postale : pour trouver un domicile de façon certaine, on commence par chercher le pays, puis la ville, puis la rue et enfin le numéro. Ici le pays est un nom de domaine de haut niveau. Ils sont bien connus de tous les internautes : .com, .net, .fr ... Le nom de sous domaine peut-être assimilé au nom de la ville dans ce pays. Il peut y avoir plusieurs villes dans le monde ayant le même nom, mais dans un même pays un nom de ville doit être unique, pour pouvoir l'identifier de façon certaine. Eventuellement, si la ville est petite, on peut se contenter de ne préciser que le nom d'une personne : le facteur saura à coup sûr où habite la personne. Pour un sous domaine, on peut également le diviser en sous réseaux ou non.

Comme un nom d'hôte complet est ordonné de façon logique, du plus précis au plus vague, le plus simple pour hiérarchiser la recherche est de le faire sur chaque partie du nom. Chaque serveur ne connaît que les noms de ses fils (le serveur pour .com sait comment atteindre www.linux-france.com mais pas www.linux-france.org), et renvoie à la racine les requêtes qu'il ne sait résoudre. Celle-ci à son tour tente de résoudre un nom en une adresse IP en renvoyant l'adresse du serveur pouvant répondre à cette demande.



hiérarchie du DNS

La hiérarchie DNS est donc divisée en zones. Une zone représente un domaine (fr, org, linux-france.org). Une zone parente peut déléguer une zone fille à un ou plusieurs serveurs de noms, et chaque zone est gérée par un serveur maître et éventuellement plusieurs serveurs secondaires dont le contenu est recopié à partir du serveur maître.

Prenons un cas pratique : résoudre le nom d'hôte machine.division.domaine.fr.

La machine cherchant à atteindre cet hôte contacte l'un des serveurs de noms par défaut (3 au maximum) ;

Si ce serveur de noms par défaut n'arrive pas à résoudre ce nom, il contacte les serveurs de noms à la racine. Il faut donc que tout serveur de noms ait au moins la liste de tous les serveurs de noms de la racine, ainsi que leur adresses IP associées. Parmi les serveurs de noms à la racine, le premier à répondre reconnaît l'adresse comme valide, et renvoie l'adresse IP du serveur de noms capable de mieux le renseigner : celui de la zone .fr (donc des noms de domaines du type xxx.fr) ;

Le DNS local interroge alors le DNS de la zone .fr. Si ce serveur de noms n'est capable de résoudre machine.division.domaine.fr, il renvoie la liste des serveurs de noms de la zone domaine.fr ;

A son tour, un des serveur de noms de la zone domaine.fr reconnaît le suffixe division.domaine.fr, le serveur de noms de la zone division.domaine.fr, qui connaît l'adresse IP de machine.division.domaine.fr ;

Il y aura donc eu au maximum 3 interrogations pour les 3 serveurs de noms par défaut, 1 pour celui de la zone racine (zone « . »), 1 pour celui de la zone .fr, 1 pour celui de la zone domaine.fr, et 1 pour celui de la zone division.domaine.fr, soit au total 7 serveurs de noms interrogés. Chacun de ces serveurs de noms ne renvoie à chaque fois que l'adresse du DNS le plus apte à répondre.

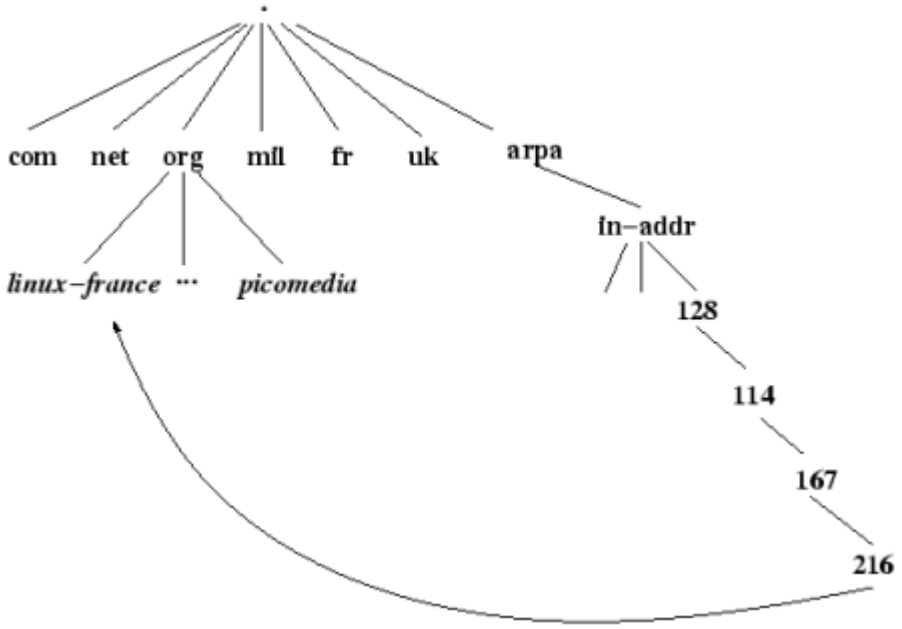
De plus, chaque serveur DNS garde en mémoire les dernières requêtes. Ainsi, si un nom est souvent demandé, il a toute les chances de figurer dans la mémoire du serveur qui n'aura pas besoin d'interroger les autres serveurs : la réponse sera directe.

On voit tout de suite l'avantage de cette méthode : au lieu d'avoir un serveur indexant toute les machines du Web, il y a des milliers de machines indexant un petit bout de l'Internet, en l'occurrence leur sous domaine. Cela répartit les informations et les charges sur ces milliers de machines. Il est donc nécessaire de configurer son propre DNS pour son réseau, si on veut que les noms des machines de son propre domaine soient résolus par d'autres hôtes.

Conversion d'adresse IP en nom

Il est parfois utile de pouvoir résoudre une adresse IP en nom de machine. Par exemple, en cas de demande de connexion de la part d'un hôte distant sur une machine, la machine distante n'envoie que son adresse IP. Il faut donc résoudre cette adresse IP en nom (cela permet au passage de vérifier que l'adresse IP est valide, et non pas « détournée » par un pirate).

Pour pouvoir résoudre une adresse en nom, un pseudo-domaine a été mis en place : le domaine in-addr.arpa.



résolution inverse en utilisant le domaine in-addr.arpa

En fait, pour l'hôte www.linux-france.org, l'adresse 128.114.167.216.in-addr.arpa. est un pointeur vers les vrais « Resource Record » de www.linux-france.org.

II] Installation du serveur DNS

a) Téléchargement des sources

Le site de l'« Internet Software Consortium » propose sa version du serveur DNS : BIND. L'adresse Internet du site est http://www.isc.org/. L'archive devrait se prénommer : « bind-latest.tar.gz ».

b) Installation du serveur BIND

La plupart des distributions Linux récentes proposent par défaut une version de BIND viable, il n'est normalement pas nécessaire de les mettre à jour.

Cependant, si vous désirez installer une nouvelle version de BIND, il vous faudra vous reportez à la documentation fourni dans l'archive précédemment téléchargée.

La décompression de cette dernière s'effectue à l'aide de la commande :

```
[root@redhat local]# tar xzvf dhcpd-latest.tar.gz
```

III] Configuration du serveur DNS

a) Serveur DNS de cache

Pourquoi ?

Cette section propose tout d'abord de configurer un serveur de noms qui ne sert que de cache, qui transmettra toute les requêtes vers le DNS du fournisseur d'accès et gardera en mémoire les réponse. Il faut mieux laisser aux DNS du fournisseur d'accès le soin d'interroger la racine pour deux raisons : la première, c'est que ces serveurs DNS ont peut être déjà en mémoire la réponse, et n'auront même pas besoin d'interroger les serveurs de noms de la racine. La seconde, c'est que ces serveurs de noms feront la demande plus rapidement que notre serveur de noms local !

Ainsi, si on consulte régulièrement la page de www.linux-france.org, la première fois le DNS local transmettra la requête au DNS du fournisseur d'accès qui lui renverra l'adresse 216.167.114.128 et la gardera en mémoire. Les requêtes suivantes pour www.linux-france.org passeront toujours par le DNS local, qui cette fois-ci aura en mémoire l'adresse 216.167.114.128, et fera lui même la requête auprès de cette adresse. Cela permettra d'accélérer les résolutions de noms en adresses IP.

La configuration

Il faut d'abord modifier deux fichiers pour indiquer quels serveurs de noms utiliser, et quels services de conversion de noms sont disponibles.

Le fichier /etc/resolv.conf détermine la façon dont le DNS doit chercher les informations :

```
# Fichier /etc/resolv.conf - Détermine la façon dont le convertisseur
# utilise le DNS. Voir les pages man Linux sous resolver(5).

# Liste des serveurs à contacter pour résoudre un nom. Il vaut mieux
# mettre en premier le serveur de noms local, pour éviter de passer par
# Internet pour une machine du réseau local. On peut mettre jusqu'a 3
# adresses.

nameserver 127.0.0.1
```

Le fichier /etc/host.conf indique quels services de conversion de noms sont disponibles, et dans quel ordre il faut les appliquer :

```
# Fichier /etc/host.conf - Indique quels services de conversion des
# noms sont disponibles, et dans quel ordre il faut les appliquer.
#
# Pour résoudre un nom en adresse IP, on peut passer soit par le
# DNS, soit par le fichier /etc/hosts. La ligne suivante indique dans
# quel ordre appliquer cette recherche : d'abord dans le fichier
# /etc/hosts, puis par le DNS en cas d'échec.
#
# Valeurs possibles : hosts, bind, nis.
order hosts, bind
```

Pour notre serveur de noms basique, on utilisera ces versions de fichiers. Pour configurer un DNS pour un autre domaine local plus tard, on y apportera des modifications.

Le premier fichier de configuration lu par named : named.conf, indique quels autres fichiers lire. Voici le fichier par défaut /etc/named.conf :

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
*/

options {
    directory "/var/named";

/*
 * Adresses des serveurs a contacter si le serveur de noms local est incapable
 * de résoudre le nom. Elles correspondent aux serveurs de noms du
 * fournisseur d'accès.
*/
    forward only;
    forwarders {
        172.16.0.4;
};

/*
 * Désignation des serveurs racines.
 */
```

```
zone "." {
    type hint;
    file "named.ca";
};

/*
 * Fichiers utilisés pour la résolution du domaine localhost.
 */

zone "0.0.127.in-addr.arpa" IN {
    type master;
    file "db.127.0.0";
};
```

Le serveur DNS va piocher les informations dans les fichiers named.ca pour la zone « . » et db.127.0.0 pour la zone 0.0.127.in-addr.arpa. Ces fichiers sont situés dans le répertoire /var/named (comme indiqué par la ligne : directory "/var/named";).

La zone « . » correspond à la zone racine (la racine à tous les domaines de l'Internet).

Le fichier /var/named/named.ca contient donc les adresses des serveurs DNS qui peuvent renseigner un utilisateur à partir de la racine de la hiérarchie.

La zone 0.0.127.in-addr.arpa permet d'effectuer la résolution inverse de toute les adresses commençant par 127.0.0, donc cette zone permet de résoudre l'adresse 127.0.0.1 en son nom.

Pour une version de bind inférieure à 8, c'est le fichier /etc/named.boot qui est lu au démarrage de named :

```
; Serveur DNS de cache
directory                               /var/named
cache           .                       named.ca
primary         0.0.127.in-addr.arpa    db.127.0.0
```

C'est à peu près les mêmes instructions que dans le fichier /etc/named.conf. Se référer au DNS HOWTO pour la suite des opérations, ou installer une version de named supérieure à 8. Pour savoir la version, tapez la commande « named -v » comme suit :

```
[root@redhat /root]# named -v
named 8.2.2-P5 Sat Aug  5 13:21:24 EDT 2000
        prospector@porky.devel.redhat.com:/usr/src/bs/BUILD/bind-8.2.2_P5/src/bin/named
```

Voici le fichier /var/named/db.127.0.0 :

```
; Pour plus d'informations, voir les pages man Linux sous named(8).
;
; Début d'autorisation (SOA=Start Of Authority). Cet enregistrement de
; ressource utilise l'adressage TCP/IP (IN). Le serveur de noms primaire
; a pour nom "localhost", la personne à contacter a pour adresse mail
; "root@localhost.".

@       IN      SOA     localhost.  root.localhost.     (
                                1       ; serial
                                10800   ; Refresh after 3 hours
                                3600    ; Retry after 1 hour
                                604800  ; Expire after one week
                                86400 ) ; Minimum TTL of 1 day
        IN      NS      localhost.
```

```
        IN      MX 10   localhost.

1       IN      PTR     localhost.
```

Le point de « localhost. » est très important. Il permet de spécifier une machine à partir de la racine et non à partir du domaine courant.

Ce domaine courant est représenté par le caractère @ et est égal à l'indication « zone » du fichier /etc/named.conf, et s'il est omis (comme à la ligne indiquant l'enregistrement NS par exemple), named considère qu'il s'agit du domaine courant précédemment indiqué.

Le fait de transmettre toute les requêtes qu'on ne peut résoudre au serveur DNS du fournisseur d'accès permet de soulager les DNS de la racine. Ainsi, au lieu de surcharger les DNS connus par le monde entier, on surcharge les DNS connus par tous les clients du fournisseur d'accès.

Voilà, le serveur de noms servant de cache est configuré, il n'y a plus qu'à le lancer pour le tester ! Pour cela, tapez :

```
[root@redhat /root]# ndc start
new pid is 3443
```

ou, si le DNS fonctionne déjà :

```
[root@redhat /root]# ndc restart
new pid is 3446
```

Pour savoir si il y a des erreurs dans les fichiers de configuration, consulter le fichier /var/log/messages. Si une erreur s'est produite, elle sera aisément visible.

Un message du type « No default TTL set using SOA minimum instead » est normal, il signifie qu'aucune zone n'étant spécifiée, il utilise la zone par défaut. En revanche, un message du type :

```
redhat named[1590]: Zone "0.0.127.in-addr.arpa" (file named.local): no NS RRs found at zone top
redhat named[1590]: master zone "0.0.127.in-addr.arpa" (IN) rejected due to errors
```

est plus grave. Il signifie qu'aucun serveur de noms n'est spécifié par l'option NS pour la zone 0.0.127.in-addr.arpa. En regardant de plus près le fichier de configuration de cette zone, on devrait se rendre compte d'un erreur de frappe.

Un bon moyen de débuguer les fichiers de configuration mal écrits est d'utiliser tail et grep :

```
[root@redhat /root]# tail -50 /var/log/messages | grep error
```

On voit tout de suite s'afficher les lignes contenant des erreurs. On peut aussi utiliser l'option -f de la commande tail pour avoir les dernières lignes du fichier, au fur et à mesure qu'elles arrivent.

Tester son serveur DNS

Pour tester le nouveau serveur de nom, on peut utiliser le programme nslookup, par exemple :

```
[root@redhat /root]# nslookup
Default Server:  localhost
Address:  127.0.0.1

> 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1

> exit
```

Conclusion : on est capable de résoudre l'adresse 127.0.0.1, donc ça marche. Si on n'a qu'une seule machine à disposition, c'est terminé pour la partie DNS. Sinon, il va falloir rendre accessible aussi la résolution des noms et/ou adresses des autres machines.

b) Un serveur DNS pour un réseau local

Dans cette section, on supposera que la machine sur laquelle tournera le serveur DNS que l'on souhaite installer s'appelle machine1 et que le domaine s'appelle domaine1. Le nom complet du serveur DNS est donc machine1.domaine1. Elle a pour adresse IP 192.168.1.1. Pour les noms d'hôtes n'appartenant pas au domaine local (comme www.linux-france.org, par exemple), on contactera le serveur DNS du fournisseur d'accès d'adresses 212.27.32.5 et 212.27.32.6 (noter que ces deux adresses sont bien des adresses de DNS d'un fournisseur d'accès, il faut donc les remplacer par les adresses de son fournisseur d'accès !).

Les autres hôtes du domaine s'appellent machine2 et machine3, d'adresses IP respectives 192.168.1.2 et 192.168.1.3.

Table : Résumé des paramètres du DNS que l'on cherche à configurer.

Nom d'hôte	Nom de domaine	Adresse IP	Adresses des serveurs DNS du FAI
machine1	domaine1	192.168.1.1	212.27.32.5
machine2	domaine1	192.168.1.2	212.27.32.6
machine3	domaine1	192.168.1.3	

Les fichiers de configuration à écrire restent les mêmes que dans le cas d'un serveur DNS ne servant que de cache. Ils sont juste un peu plus longs.

Voici le fichier /etc/named.conf :

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
*/

options {
    directory "/var/named";
```

```
/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de résoudre le nom.
 */

    forward only;
    forwarders {
        212.27.32.5;
        212.27.32.6;
    };
};

/*
 * Fichier de cache.
 */

zone "." {
    type hint;
    file "named.ca";
};

/*
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 127.0.0 peuvent être résolues en nom d'hôte dans le fichier spécifié.
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.0.0";
};

/*
 * Fichier utilise pour la résolution des noms d'hôte se terminant par
 * domaine1
 */

zone "domaine1" {
    type master;
    file "db.domaine1";
};

/*
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 192.168.1 peuvent être résolues en nom d'hôte dans le fichier
 * spécifié.
 */

zone "1.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.1";
};
```

Le fichier /var/named/db.127.0.0 reste presque le même que pour le cas d'un serveur ne servant que de cache. La ligne indiquant le serveur de noms pour la zone change :

```
; Début d'autorisation (SOA=Start Of Authority). Cet enregistrement de
; ressources utilise l'adressage TCP/IP (IN). Le serveur de noms primaire
; a pour nom "machine1.domaine1.", la personne a contacter a pour
; adresse mail "root@machine1.domaine1".

@    IN    SOA    machine1.domaine1.    root.machine1.domaine1. (

; Les options qui suivent n'ont pas a être modifiées dans le cas d'une
```

```
; configuration simple. Elles concernent le serveur de noms secondaire.

    1997022700 ; serial
    28800      ; refresh
    14400      ; retry
    3600000    ; expire
    86400      ; default_ttl
)

; "machine1.domaine1." est le serveur de noms pour le domaine.

     IN    NS    machine1.domaine1.

; l'adresse domaine_local.1 (soit 192.168.1.1) sera associée au nom
; "localhost.".

1    IN    PTR    localhost.
```

Voici le fichier /var/named/db.domaine1 :

```
; Fichier /var/named/db.domaine1 - Contient les informations sur
; la zone domaine1 pour résoudre un nom d'hôte en adresse IP.
;
; Le domaine local, représenté par le caractère ``@'', est donc
; "domaine1".

@    IN    SOA    machine1.domaine1.    root.machine1.domaine1. (
    2000070306 ; serial
    3600       ; refresh
    900        ; retry
    1209600    ; expire
    43200      ; default_ttl
)

; Permet d'associer le domaine local (@) a l'adresse IP 192.168.1.1.

     IN    A    192.168.1.1

; Ce qui suit est assez explicite...

     TXT    "Serveur DNS local de domaine1"

; "machine1" et "machine1.domaine1" sont les serveurs de noms pour
; le domaine1.

@    IN     NS    machine1
@    IN     NS    machine1.domaine1.

; Echange de mail : tout mail doit être envoyé d'abord a machine1, puis a
; machine1.domaine1. Les nombres indiquent la priorité (la valeur la plus
; faible correspond à la priorité la plus importante).

@    IN    MX    10    machine1
@    IN    MX    20    machine1.domaine1.

; Informations concernant chaque machine du réseau devant être résolue
; par le serveur de noms local.

; Adresse de référence de machine1 : 192.168.1.1 (type A, unique pour une
; adresse). Diverses informations sur la machine peuvent être spécifiées
; par HINFO. Ici, c'est en commentaires car problème sécurité ;-)
;
```

```
machine1    IN    A        192.168.1.1
machine1.   IN    A        192.168.1.1
machine1    IN    HINFO    "Intel P133+" "Linux 2.2.14-12"

; Les autres machines du réseau

machine2    IN    A        192.168.1.2
machine3    IN    A        192.168.1.3

; localhost, pour l'interface de bouclage.

localhost   IN    A        127.0.0.1

; Quelque alias (CNAME) : un nom d'hôte précisé par un enregistrement de
; type A peut avoir un ou plusieurs alias. Dans l'exemple ci-dessous,
; www => machine1, ftp => machine1, mail => machine1, si bien que
; www.domaine1, ftp.domaine1 ou encore mail.domaine1
; désignent la même adresse IP, celle de l'enregistrement de type A de
; machine1. On peut ainsi faire une requête du style :
;
;     - lynx http://www.domaine1
;     - ftp ftp.domaine1
;     - fetchmail mail.domaine1

www     CNAME    machine1
ftp     CNAME    machine1
mail    CNAME    machine1
```

Enfin, voici le fichier /var/named/db.192.168.1 :

```
; Fichier /var/named/db.192.168.1 - Contient les informations sur
; la zone domaine1 pour résoudre une adresse IP en nom d'hôte.
;
; Le domaine local, represente par le caractere ``@'', est donc
; "1.168.192.in-addr.arpa".

@    IN    SOA    machine1.domaine1.    root.machine1.domaine1. (
    2000070305 ; serial
    3600 ; refresh
    900 ; retry
    1209600 ; expire
    43200 ; default_ttl
)

; Serveur de noms pour le domaine.

    IN    NS    machine1.domaine1.

; Associe des adresses IP a des noms. Les adresses sont données par rapport
; au domaine local. Ainsi, 2 <=> 2.1.168.192.in-addr.arpa.

1    IN    PTR    machine1.domaine1.
2    IN    PTR    machine2.domaine1.
3    IN    PTR    machine3.domaine1.
```

Tester le nouveau réseau

Comme précédement, on peut tester notre nouvelle configuration avec nslookup :

```
[root@redhat /root]# nslookup
Default Server:  machine1.domaine1
Address:  192.168.1.1

> machine1.domaine1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine1
Address:  192.168.1.1

> machine2.domaine1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine2.domaine1
Address:  192.168.1.2

> 192.168.1.1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine1
Address:  192.168.1.1

> 192.168.1.3
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine3.domaine1
Address:  192.168.1.3

> www.domaine1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine1
Address:  192.168.1.1
Aliases:  www.domaine1

> exit
```

c) Un serveur DNS secondaire

Pour équilibrer les charges des DNS locaux, on peut vouloir mettre en place plusieurs DNS pour notre domaine local. Dans ce cas on met d'abord en place un DNS primaire qui servira de « maître » pour le DNS secondaire. Dans les sections précédentes, on a précisé que le DNS que l'on mettait en place était « maître » pour le domaine local comme l'indique la ligne type master; du fichier /etc/named.conf.

Dans notre cas, le fichier /etc/named.conf du DNS secondaire doit préciser que c'est un serveur « esclave », et préciser l'adresse IP de son serveur « maître » :

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
*/

options {
    directory "/var/named";

/*
 * Désactive l'envoi de message aux serveurs esclaves pour leur indiquer des
 * modifications de zone.
 */

    notify no;
/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de résoudre le nom. Cette option peut remplacer la commande nameserver
 * du fichier /etc/resolv.conf.
 */

    forward only;
    forwarders {
        212.27.32.5;
        212.27.32.6;
    };
};

/*
 * Fichier de cache.
 */

zone "." {
    type hint;
    file "named.ca";
};

/*
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 127.0.0 peuvent être résolues en nom d'hôte dans le fichier spécifié
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.0.0";
};
```

```
/*
 * Fichier utilisé pour la résolution des noms d'hôte se terminant par
 * domaine1. C'est un DNS secondaire, qui est mis a jour
 * automatiquement d'après les informations du DNS sur la machine
 * d'adresse(s) IP :
 *     192.168.1.1
 */

zone "domaine1" {
    type slave;
    file "db.domaine1";
    masters { 192.168.1.1; };
};

/*
 * Fichier utilisé pour la résolution inverse. Les adresses IP commençant
 * par 192.168.1 peuvent être résolues en nom d'hôte dans le fichier
 * spécifié
 */

zone "1.168.192.in-addr.arpa" {
    type slave;
    file "db.192.168.1";
    masters { 192.168.1.1; };
};
```

On peut préciser plusieurs serveurs maîtres, séparés par un point-virgule (« ; »).

La mise à jour du DNS « esclave » est automatique à partir du DNS « maître ». Ainsi, insérer une nouvelle machine dans le DNS prend peu de temps (il n'y a que quelque lignes à rajouter dans deux fichiers), et tous les serveurs secondaires seront mis à jour automatiquement !

Par défaut, à chaque changement de numéro de série du serveur maître, un message est envoyé aux serveurs esclaves pour leur indiquer qu'il faut mettre à jour leur configuration. L'option notify permet de désactiver cela. On peut par la suite la réactiver ou non pour chaque zone.

Les options figurant au début du fichier de zone (serial, refresh...) servent pour le DNS secondaire. Détaillons ces options :

serial
C'est le numéro (un nombre entier) de version du fichier d'information de zones. Ce numéro est utilisé par les DNS secondaires pour savoir si le fichier d'informations de zone du DNS primaire a été changé. Il doit être augmenté de 1 à chaque modification du fichier.

refresh
Intervalle de temps en secondes durant lequel le DNS secondaire attend avant de vérifier (et éventuellement mettre à jour) l'enregistrement SOA du DNS primaire. Ces enregistrements ne changent pas souvent en général, une journée (86400 secondes) peut largement suffire.

retry
Intervalle de temps en secondes durant lequel le DNS secondaire attend avant de réessayer une requête vers le DNS primaire si celui ci n'est pas accessible. Cette valeur devrait être de quelque minutes.

expire
Intervalle de temps en secondes durant lequel le DNS secondaire attend avant de rejeter les informations de zones si il n'a pu contacter le DNS primaire. Cette valeur devrait être de plusieurs jours (voir plusieurs mois).

d) Configurer un serveur DNS derrière un firewall

Si il y a un firewall entre le DNS que l'on veut configurer et les autres DNS à contacter, il faut rajouter la ligne query-source address * port 53; dans le fichier /etc/named.conf.

Voici le début du nouveau fichier /etc/named.conf :

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
*/

options {
    directory "/var/named";

/*
 * On est derrière un firewall
 */

    query-source address * port 53;

/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de résoudre le nom. Cette option peut remplacer la commande nameserver
 * du fichier /etc/resolv.conf.
 */

    forward only;
    forwarders {
        212.27.32.5;
        212.27.32.6;
    };
};

...
```

e) Configurer plusieurs domaines

On peut configurer plusieurs domaines pour un même serveur DNS. Cela peut être intéressant par exemple si le réseau est divisé en sous-domaines. Dans ce cas, il faut définir deux zones par nom de domaine, avec un nouveau fichier associé par zone (un pour la résolution directe et un pour la résolution inverse).

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
*/

options {
    directory "/var/named";

/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de résoudre le nom. Cette option peut remplacer la commande nameserver
 * du fichier /etc/resolv.conf.
 */
```

```
    forward only;
    forwarders {
        212.27.32.5;
        212.27.32.6;
    };

};

/*
 * Fichier de cache.
 */

zone "." {
    type hint;
    file "named.ca";
};

/*
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 127.0.0 peuvent être résolues en nom d'hôte dans le fichier spécifié
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "db.127.0.0";
};

/*
 * Fichier utilise pour la résolution des noms d'hôte se terminant par
 * domaine1
 */

zone "domaine1" {
    type master;
    file "db.domaine1";
};

/*
 * Fichier utilise pour la resolution inverse. Les adresses IP commencant
 * par 192.168.1 peuvent etre resolues en nom d'hote dans le fichier
 * specifie
 */

zone "1.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.1 ";
};

/*
 * Autres domaines que le domaine local. On pourrait mettre tous les sous
 * domaines du domaine local, par exemple
 */

/*
 * Fichier utilise pour la résolution des noms d'hôte se terminant par
 * domaine2 (domaine 192.168.2)
 */

zone "domaine2" {
    type master;
    file "db.domaine2";
};

/*
```

```
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 192.168.2 peuvent être résolues en nom d'hôte dans le fichier
 * spécifié.
 */

zone "2.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.2";
};

/*
 * Fichier utilise pour la résolution des noms d'hôte se terminant par
 * domaine3 (domaine 192.168.3)
 */

zone "domaine3" {
    type master;
    file "db.domaine3";
};

/*
 * Fichier utilise pour la résolution inverse. Les adresses IP commençant
 * par 192.168.3 peuvent être résolues en nom d'hôte dans le fichier
 * spécifié
 */

zone "3.168.192.in-addr.arpa" {
    type master;
    file "db.192.168.3";
};
```

Tester notre nouvelle configuration

Une fois encore, on teste notre nouvelle configuration avec nslookup :

```
[root@redhat /root]# nslookup
Default Server:  machine1.domaine1
Address:  192.168.1.1

> machine1.domaine1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine1
Address:  192.168.1.1

> machine1.domaine2
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine2
Address:  192.168.2.1

> machine2.domaine3
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine2.domaine3
Address:  192.168.3.2


> 192.168.2.3
```

```
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine3.domaine2
Address:  192.168.2.3

> set q=any
> machine2.domaine3
Server:  machine1.domaine1
Address:  192.168.1.1

machine2.domaine3       CPU = Non definit       OS = Non definit
machine2.domaine3       internet address = 192.168.3.2
domaine3        nameserver = machine1.domaine1
machine1.domaine1       internet address = 192.168.1.1
```

f) Déléguer une zone d'un sous-domaine

Pour des grands domaines, il peut être utile de mettre en place plusieurs serveurs serveurs DNS, chacun gérant sa zone correspondant à son sous-domaine.

Si le domaine domaine1.fr veut déléguer la gestion des sous-domaines division1.domaine1.fr, division2.domaine1.fr...aux serveurs de noms ns.division1.domaine1.fr (192.168.1.1) et ns.division2.domaine1.fr (192.168.2.1), il faut que dans le fichier de zone de domaine1.fr figurent les lignes suivantes :

```
; Délégation des sous domaines division1.domaine1.fr et division2.domaine1.fr

division1.domaine1.fr.       IN    NS    ns.division1.domaine1.fr.
division2.domaine1.fr.       IN    NS    ns.division2.domaine1.fr.

ns.division1.domaine1.fr.    IN    A     192.168.1.1
ns.division2.domaine1.fr.    IN    A     192.168.2.1
```

Pour la résolution inverse, il faut compléter le fichier de résolution inverse db.domaine1.fr.rev comme suit :

```
; Délégation des sous domaines division1.domaine1.fr et division2.domaine1.fr
;

1.168.192.in-addr.arpa.    IN    NS    ns.division1.domaine1.fr.
2.168.192.in-addr.arpa.    IN    NS    ns.division2.domaine1.fr.
```

III] Débuguer un serveur DNS

Il est admis que l'écriture des fichiers de configuration d'un DNS est difficile. En plus de cela, il faut faire très attention à ce que l'on écrit, un espace ou un point oubliés peuvent tout changer, et pour trouver l'erreur, on peut y passer des heures ! Cette section présente quelques trucs et sources d'erreurs possibles.

Consulter le fichier /var/log/messages après chaque relancement de named. Les erreurs apparaîtront avec le nom du fichier et la ligne incriminée ;

Vérifier les noms des hôtes dans les fichiers de configuration. Ne pas oublier qu'un nom est relatif à la zone si il ne se termine pas par un point ;

Pour un DNS secondaire, ne pas oublier d'incrémenter le numéro de série dans les fichiers de configuration du DNS primaire pour qu'ils soient pris en compte ;

Ne pas oublier que si un champ est facultatif, il faut quand même laisser un espace (IN MX 10 machine et MX 10 machine sont équivalents, mais il doit y avoir un espace avant MX) ;

Si une zone extérieure semble ne pas être atteignable, utiliser l'option debug de nslookup :

```
[root@redhat /]# nslookup
Default Server:  machine1.domaine1
Address:  192.168.1.1

> set debug
> www.linux-france.org
Server:  intranet.linagora.com
Address:  192.168.0.3

;; res_nmkquery(QUERY, www.linux-france.org, IN, A)
------------
Got answer:
    HEADER:
        opcode = QUERY, id = 10124, rcode = NOERROR
        header flags:  response, want recursion, recursion avail.
        questions = 1,  answers = 2,  authority records = 2,  additional = 2

    QUESTIONS:
        www.linux-france.org, type = A, class = IN
    ANSWERS:
    ->  www.linux-france.org
        canonical name = linux-france.org
        ttl = 1495 (24m55s)
    ->  linux-france.org
        internet address = 216.167.114.128
        ttl = 1069 (17m49s)
    AUTHORITY RECORDS:
    ->  linux-france.org
        nameserver = NS1.SLCONSEIL.COM
        ttl = 144360 (1d16h6m)
    ->  linux-france.org
        nameserver = MERCEDES.NFRANCE.COM
        ttl = 144360 (1d16h6m)
    ADDITIONAL RECORDS:
    ->  NS1.SLCONSEIL.COM
        internet address = 216.167.108.244
        ttl = 143351 (1d15h49m11s)
    ->  MERCEDES.NFRANCE.COM
        internet address = 212.208.53.2
        ttl = 135665 (1d13h41m5s)

------------
Non-authoritative answer:
Name:    linux-france.org
Address:  216.167.114.128
Aliases:  www.linux-france.org
```

III] DNS et sécurité

Il est possible (et même souhaitable) de sécuriser son DNS local. Voici quelque recettes. Pour plus de détails, se reporter au livre « Securing and Optimizing Linux : Red Hat Edition [2] ».

Dans le fichier /etc/named.conf, on peut spécifier les DNS autorisés à demander un transfert de zone à l'aide de l'option allow-transfer :

```
/*
 * Fichier utilisé pour la résolution des noms d'hôte se terminant par
 * domaine1. Seul le DNS d'adresse 192.168.2.1 a le droit de récupérer les
 * informations a partir de ce DNS.
 */

zone "domaine1" {
    type master;
    file "db.domaine1";
    allow-transfer { 192.168.2.1; };
};
```

Les transferts de zones étant utilisés par les spammers et les spoofers d'IP, il est recommandé de spécifier cette option. Si on n'a pas de DNS secondaire, on peut mettre l'adresse loopback (127.0.0.1). On peut préciser plusieurs adresses, séparées par un point-virgule ;

Vérifier que chaque adresse IP faisant une requête au DNS est bien associée à un nom de domaine valide à l'aide de l'option nospoof dans le fichier /etc/host.conf. On peut également enregistrer chaque tentative de spoofing à l'aide de syslog avec l'option alert ;

Limiter les interfaces sur lesquelles named tourne en mettant listen-on { 192.168.1.1 ; } ; dans le fichier /etc/named.conf, où 192.168.1.1 est l'adresse du serveur DNS ;

Autoriser les requêtes au DNS de la part des hôtes d'un domaine particulier, les autres n'y étant pas autorisés. Par exemple, pour que seuls les hôtes du domaine local 192.168.1.0 soient autorisés à interroger le DNS, insérer dans le fichier /etc/named.conf l'option allow-query :

```
/*
 * Fichier utilise pour la résolution des noms d'hôte se terminant par
 * domaine1. Seul le DNS d'adresse 192.168.2.1 a le droit de récupérer les
 * informations a partir de ce DNS. Seuls les hôtes du domaine 192.168.1.0/24
 * sont autorisés a interroger ce DNS local.
 */

zone "domaine1" {
    type master;
    file "domaine1";
    allow-transfer { 192.168.2.1; };
    allow-query {192.168.1.0/24; };
};
```

L'adresse 192.168.1.0/24 signifie « toute les adresses dont les 24 premiers bits commencent par 192.168.1.0 ». Comme on a une adresse de classe C avec un masque de réseau correspondant à une adresse de classe C, cela revient à dire tous les hôtes du réseau 192.168.1.0 ;

Empêcher un utilisateur de déterminer la version de BIND :

```
zone "bind" chaos {
    type master;
    file "bind";
    allow-query { localhost ; };
};
```

Le fichier bind contient :

```
$TTL 1d
@    CHAOS    SOA    localhost. root.localhost. (
     1   ; serial
     3H  ; refresh
     15M ; retry
     1W  ; expire
     1D  ; minimum
)

    NS    localhost.
```

Faire tourner le DNS sous l'identité d'un utilisateur normal, utilisé uniquement pour les besoins du DNS :

```
[root@redhat /]# useradd -M -r -d /var/named -s /bin/false named
[root@redhat /]# groupadd -r named
```

Ne pas oublier de changer le script d'initialisation (/etc/rc.d/init.d/named sous RedHat ou /etc/init.d/named sous Debian) pour y mettre la ligne :

```
/usr/sbin/named -u named -g named
```

Ne pas mettre de RR de type HINFO. Les informations associées donnant des informations sur la machine sur laquelle tourne le DNS, on peut plus facilement trouver les failles de sécurité ;

Autoriser les modifications selon une clé

```
...
/*
 * Désignation de la clé ainsi que de son algorithme de cryptage
 */

key DHCP_UPDATER {
      algorithm HMAC-MD5.SIG-ALG.REG.INT;
      secret pRP5FapFoJ95JEL06sv4PQ==;
}

/*
 * Déclarations des diverses zones modifiables
 */

zone "100.16.172.in-addr.arpa" IN {
      type master;
      file "db.172.16.100";
      allow-update { key DHCP_UPDATER; };
};
...
```