Configurer un DNS

© Mathieu DECORE

13 décembre 2000

Table des matières

1 Configurer un serveur de noms (DNS)

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[1], 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 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.

Dans la suite de ce document, le terme DNS (Domaine Name Server, serveur de noms en français) fait référence au serveur DNS.

Pour résoudre un nom en adresse IP, la méthode la plus simple consiste à mettre tous les noms d'hôtes et leurs adresses associées dans le fichier `/etc/hosts` :

```
127.0.0.1       localhost.localdomain       localhost
192.168.1.1     zecastor.athome.chezmoi     zecastor
192.168.1.2     nougat.athome.chezmoi       nougat
```

Cette méthode peut se révéler fastidieuse à la longue : chaque fois qu'on veut insérer une nouvelle machine dans le réseau, il faut modifier le fichier `/etc/hosts` de **chaque** machine. De plus, ces machines seront difficilement identifiables par des machines non Unixiennes (comme Windows, par exemple). Enfin, un DNS rend plus facile la distribution des méls dans un réseau local et permet de mettre plus facilement en place des alias sur des noms, et offre la mémorisation des adresses résolues par un autre DNS (comme celui du fournisseur d'accès, par exemple).

Note : la mise en place d'un serveur DNS rend obsolète le fichier `/etc/hosts`. Il est néanmoins conseillé de continuer à le mettre à jour.

1.1 Le DNS : la théorie...

Avant d'entrer dans le vif du sujet, cette section propose d'expliquer le principe de fonctionnement du DNS. Pour plus de détails, consulter [4] [5]. Comme on l'a vu, pour des raisons de commodité, il est plus facile pour un humain de manipuler des noms significatifs, tels que **www.linux-france.org**, et des adresses codées sur plusieurs octets pour une machine (comme *216.167.114.128*).

1. En fait, une machine peut avoir plusieurs adresses IP.

Au début des années 1970, un système centralisé à Stanford 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, Systeme de Nom de Domaine en français) a été conçu pour résoudre ce problème, en proposant un modèle hiérarchisé.

Chaque machine (imprimante, 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 resaux 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 requettes qu'il ne sait résoudre. Celle-ci à son tour tente de résoudre une adresse IP en nom en renvoyant l'adresse du serveur pouvant répondre à cette demande (figure 1).

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.

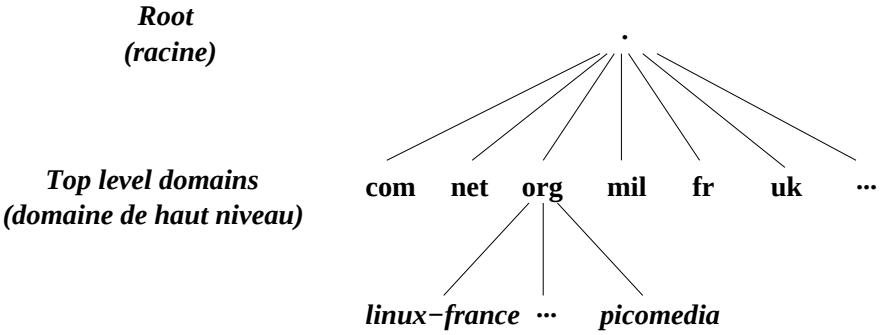

FIG. 1 – *Hiérarchie du DNS*

1.1.1 Conversion de nom en adresse IP

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

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

2. Si ce serveur de noms par défaut n'arrive pas à resoudre 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. Parmis 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**) ;

3. 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**;

4. 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 requettes. 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'occurence 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'autes hôtes.

1.1.2 Conversion d'adresse IP en nom

Il est parfois utile de pouvoir résoudre un nom en adresse IP. Par exemple, en cas de connexion de la part d'un hôte distant sur une machine, le nom de la machine distante est cherchée dans un fichier tel que `/etc/host.equiv` pour une éventuelle connexion automatique. Hors, au moment de la demande de connexion, 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 resoudre une adresse en nom, un pseudo-domaine a été mis en place : le domaine **in-addr.arpa.** (voir figure 2).

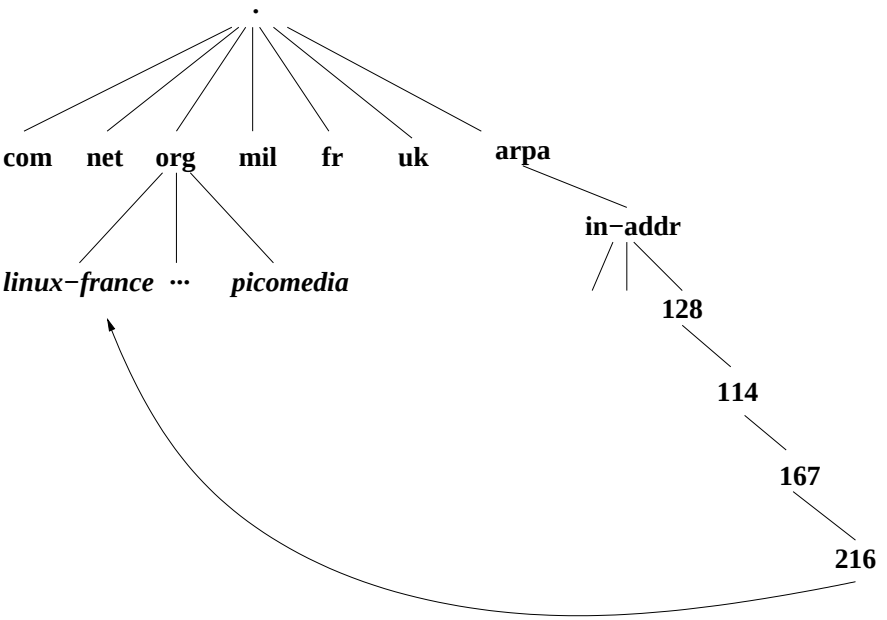

FIG. 2 – *Résolution inverse en utilisant le domaine in-addr.arpa*

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**.

1.1.3 Le resolver

Pour interroger un serveur de noms, on n'a pas besoin d'avoir de DNS local. Pour toute requête, on passe par une bibliothèque appelée "resolver". Il s'agit d'un ensemble de fonctions écrites en C (comme *gethostbyname*, par exemple). Cette bibliothèque est indépendante du serveur DNS. Ainsi, on peut se contenter d'interroger des serveurs de noms extérieurs sans avoir de DNS local. On verra dans la section 1.2 qu'il faut mieux quand même configurer un DNS local dans ce cas.

Le fichier de configuration du resolver s'appelle `/etc/resolv.conf` :

```
#
# Fichier /etc/resolv.conf - Determine la facon dont le convertisseur
# utilise le DNS. Voir resolver(5).
#
# Indique le nom de domaine local. Si cette option n'est pas specifiee,
# utilise l'appel systeme getdomainname (voir getdomainname(2)).
# Cherche pour le domaine entier, puis sur le domaine pere en cas d'echec.
#
# Par exemple :
#
#     - nougat   -> nougat                   => succes (figure dans le
#                                               fichier /etc/hosts)
#
#     - zecastor -> zecastor                 => echec
#                -> zecastor.athome.chezmoi  => succes
#
#     - alsace   -> alsace                   => echec
#                -> alsace.athome.chezmoi    => echec
#                -> alsace.chezmoi           => echec
#

domain athome.chezmoi

#
# Liste des domaines a chercher, dans l'ordre.
#
# Par exemple :
#
#     - zecastor -> zecastor.athome.chezmoi  => succes
```

```
#
#     - alsace   -> alsace.athome.chezmoi    => echec
#                -> alsace.ens.uvsq.fr       => succes

search athome.chezmoi ens.uvsq.fr wapmove.com

#
# Liste des serveurs a contacter pour resoudre un nom. Il faut mieux
# mettre en premier le serveur de noms local, pour eviter de passer par
# internet pour une machine du reseau local. On peut mettre jusqu'a 3
# adresses.
#

nameserver 192.168.1.1
nameserver 212.27.32.5
```

1.2 ... et la pratique

Cette section propose tout d'abord de configurer un serveur de noms qui ne sert que de cache, qui transmettra toute les requettes 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 requette au DNS du fournisseur d'accès qui lui renverra l'adresse *216.167.114.128* et la gardera en mémoire. Les requettes 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 requette auprès de cette adresse. Cela permettra d'accelerer les resolutions de noms en adresses IP. Ensuite, on verra la configuration d'un DNS pour notre domaine local, en y insérant toutes les machines du domaine, des alias, ... Enfin, on verra comment mettre en place un serveur de noms secondaire.

1.3 Les fichiers de 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 informa-

tions :

```
#
# Fichier /etc/resolv.conf - Determine la facon dont le convertisseur
# utilise le DNS. Voir resolver(5).
#

#
# Liste des serveurs a contacter pour resoudre un nom. Il faut mieux
# mettre en premier le serveur de noms local, pour eviter de passer par
# internet pour une machine du reseau 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 resoudre 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'echec.
#
# 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 apportera des modifications à ces fichiers.

Pour la suite, il faut avoir installé les packages `caching-nameserver-*` et `bind-*`.

1.4 Un serveur de noms qui ne sert que de cache

Le premier fichier de configuration lu par `named` 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)
 *
 * Specifie le(s) chemin(s) ou se trouvent les fichiers de zone DNS.
 *
 * Valeurs possibles : directory chemin
 */

options {
    directory "/var/named";

/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de resoudre 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 resolution inverse. Les adresses IP commencant
 * par 127.0.0 peuvent etre resolues en nom d'hote dans le fichier specifie
 */
```

```
zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};
```

Le DNS va piocher les informations dans les fichiers `named.ca` pour la zone `.` et `named.local` 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, voir figure 1).

Le fichier `/var/named/named.ca` contient donc l'adresse des 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 commencant par **127.0.0**, donc cette zone permet de résoudre l'adresse **127.0.0.1** en son nom.

On configure ici un serveur de noms **maître** comme l'indique la ligne `type master;`. On verra à la section 1.6 comment configurer un serveur de noms esclave.

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

```
;
; a caching only nameserver config
;
directory                               /var/named
cache           .                       named.ca
primary         0.0.127.in-addr.arpa    named.local
```

C'est à peu près les mêmes instructions que dans le fichier `/etc/named.conf`. Se référer à la DNS HOW-TO [1] pour la suite des opérations, ou installer `named` version supérieure à 8 (plus sécurisée). Pour savoir la version, taper `named -v` :

```
# named -v
named 8.2.2-P5 Mon Feb 28 10:17:53 EST 2000
        root@porky.devel.redhat.com:/usr/src/bs/BUILD/bind-8.2.2_P5/src/bin/named
#
```

Voici le fichier `/var/named/named.local` :

```
;
; Enregistrement de ressources. Syntaxe :
;
;     proprietaire ttl classe type donnees
;
; ou :
;
;     - proprietaire : nom de domaine ou nom d'hote relatif a
;     l'enregistrement. Nom de domaine de l'enregistrement de ressources
;     precedent par defaut ;
;
;     - ttl (time to live) : indique la duree de validite des informations,
;     en secondes, depuis leur recuperation depuis un serveur DNS. ttl
;     minimal du dernier debut d'autorisation par defaut ;
;
;    - classe : classe d'adresse du reseau. IN pour les resaux TCP/IP (ceux
;    qu'on utilise). Celle de l'enregistrement de ressources precedent par
;    defaut ;
;
;    - type : type d'enregistrement de ressources. Champ OBLIGATOIRE. Le
;    type sera explicite dans chaque exemple ;
;
;    - donnees : donnees associes au type d'enregistrement ;
;
; Cf. named(8).
;
; Debut d'autorisation (SOA=Start Of Authority). Cet enregistrement de
; ressources utilise l'adressage TCP/IP (IN). Le serveur de noms primaire
; a pour nom "localhost", la personne a contacter a pour adresse mel
; "root@localhost".
;

@       IN      SOA     localhost. root.localhost.  (

;
; Les options qui suivent n'ont pas a etre modifiees dans le cas d'une
; configuration simple. Elles concernent le serveur de noms secondaire.
;


    1997022700 ; Serial
    28800      ; Refresh
```

```
    14400       ; Retry
    3600000     ; Expire
    86400 )     ; Minimum

;
; "localhost" est le serveur de noms pour le domaine.
;


        IN      NS      localhost.

;
; l'adresse domaine_local.1 (soit 127.0.0.1) sera accociee au nom
; "localhost".
;

1       IN      PTR     localhost.
```

Noter qu'un nom d'hôte seul se termine par un point (".") si c'est un nom relatif au domaine courant (le domaine spécifié par la ligne `zone "0.0.127.in-addr.arpa" {` dans le fichier `/etc/named.conf`). Ce point est donc **extremement important** ! Ce domaine courant est représenté par le caractère `@`, et si il est homis (comme à la ligne indiquant le serveur de noms pour le domaine, par exemple), `named` considère qu'il s'agit du domaine courant.

Le fait de transmettre toute les requêtes qu'on ne peut résoudre au 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 relancer pour tester ! Pour cela, taper :

```
# ndc start
new pid is 3443
#
```

ou, si le DNS tourne déjà :

```
# 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 tout s'est bien passé, on devrait avoir :

```
Sep 28 19:26:02 zecastor named[1513]: named shutting down
Sep 28 19:26:02 zecastor named[1513]: USAGE 970161962 970161765 CPU=0.01u/0s
CHILDCPU=0u/0s
Sep 28 19:26:02 zecastor named[1513]: NSTATS 970161962 970161765
Sep 28 19:26:02 zecastor named[1513]: XSTATS 970161962 970161765 RR=0 RNXD=0
RFwdR=0 RDupR=0 RFail=0 RFErr=0 RErr=0 RAXFR=0 RLame=0 ROpts=0 SSysQ=1 SAns=0
SFwdQ=0 SDupQ=36 SErr=0 RQ=0 RIQ=0 RFwdQ=0 RDupQ=0 RTCP=0 SFwdR=0 SFail=0
SFErr=0 SNaAns=0 SNXD=0
sep 28 19:26:03 zecastor named: named shutdown succeeded
Sep 28 19:26:03 zecastor named[1556]: starting.  named 8.2.2-P5 Mon Feb 28
10:17:53 EST 2000 ^Iroot@porky.devel.redhat.com:/usr/src/bs/BUILD/bind-8.2.2_
P5/src/bin/named
Sep 28 19:26:03 zecastor named[1556]: hint zone "" (IN) loaded (serial 0)
Sep 28 19:26:03 zecastor named[1556]: Zone "0.0.127.in-addr.arpa" (file
named.local): No default TTL set using SOA minimum instead
Sep 28 19:26:03 zecastor named[1556]: master zone "0.0.127.in-addr.arpa" (IN)
loaded (serial 1997022700)
Sep 28 19:26:03 zecastor named[1556]: listening on [127.0.0.1].53 (lo)
Sep 28 19:26:03 zecastor named[1556]: listening on [192.168.1.1].53 (eth0)
Sep 28 19:26:03 zecastor named[1556]: Forwarding source address is [0.0.0.0].
1033
Sep 28 19:26:03 zecastor named[1557]: group = 25
Sep 28 19:26:03 zecastor named[1557]: user = named
Sep 28 19:26:03 zecastor named[1557]: Ready to answer queries.
sep 28 19:26:03 zecastor named: named startup succeeded
```

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

```
Sep 28 19:27:49 zecastor named[1590]: Zone "0.0.127.in-addr.arpa" (file
named.local): no NS RRs found at zone top
Sep 28 19:27:49 zecastor named[1590]: master zone "0.0.127.in-addr.arpa"
(IN) rejected due to errors (serial 1997022700)
```

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 près le fichier de configuration de cette zone, on se rend compte qu'il manque un espace au début de la ligne suivante :

```
IN      NS      localhost.
```

En effet, ce n'est pas parce qu'on peut se passer de spécifier un domaine qu'il ne faut pas laisser d'espace ! En rajoutant un espace au début de la ligne, le problème est corrigé.

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

```
# tail -50 /var/log/messages | grep error
Sep 28 19:27:49 zecastor named[1590]: master zone "0.0.127.in-addr.arpa"
(IN) rejected due to errors (serial 1997022700)
#
```

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.

Pour tester le nouveau serveur de noms installé, on peut utiliser le programme `nslookup`[2], par exemple :

```
# nslookup 127.0.0.1
Server:  localhost
Address:  127.0.0.1

Name:    localhost
Address:  127.0.0.1

#
```

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.

1.5 Un DNS pour un réseau local

Dans la suite de cet article, 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**. Il 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 *212.27.32.5* (noter que ces deux adresses sont bien des

2. contenu dans le package dnsutils pour Debian, ou bind-utils pour Red Hat.

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*.

Le tableau 1 résume tout cela :

Nom d'hôte	**Nom de domaine**	**Adresse IP**	**Adresses du DNS du fournisseur d'accès**
machine1	domaine1	192.168.1.1	212.27.32.5 212.27.32.5
machine2	domaine1	192.168.1.2	-
machine3	domaine1	192.168.1.3	-

TAB. 1 – *Résumé des paramètres du DNS que l'on cherche à configurer.*

Il faudra bien sûr préciser que toute les nouvelles requêtes se feront à partir du DNS d'adresse *192.168.1.1* dans le fichier `/etc/resolv.conf` en mettant dans ce fichier la ligne `nameserver 192.168.1.1` en premier.

Noter par ailleurs que les adresses de DNS du fournisseur d'accès pourraient figurer dans ce même fichier `/etc/resolv.conf` (on peut en spécifier jusqu'à **3**), mais cette méthode est désuette. Il faut mieux maintenant le spécifier dans le fichier `/etc/named.conf` par l'instruction *forwarders* comme c'est indiqué dans la suite.

1.5.1 Fichiers à écrire

Les fichiers 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)
 *
 * Specifie le(s) chemin(s) ou se trouvent les fichiers de zone DNS.
 *
 * Valeurs possibles : directory chemin
 */
```

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

/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de resoudre 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 resolution inverse. Les adresses IP commencant
 * par 127.0.0 peuvent etre resolues en nom d'hote dans le fichier specifie
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};

/*
 * Fichier utilise pour la resolution des noms d'hote se terminant par
 * domaine1
 */

zone "domaine1" {
    type master;
    file "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 "domaine1.rev";
};
```

Le fichier `/var/named/named.local` 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 :

```
;
; Enregistrement de ressources. Syntaxe :
;
;     proprietaire ttl classe type donnees
;
; ou :
;
;     - proprietaire : nom de domaine ou nom d'hote relatif a
;     l'enregistrement. Nom de domaine de l'enregistrement de ressources
;     precedent par defaut ;
;
;     - ttl (time to live) : indique la duree de validite des informations,
;     en secondes, depuis leur recuperation depuis un serveur DNS. ttl
;     minimal du dernier debut d'autorisation par defaut ;
;
;    - classe : classe d'adresse du reseau. IN pour les resaux TCP/IP (ceux
;    qu'on utilise). Celle de l'enregistrement de ressources precedent par
;    defaut ;
;
;    - type : type d'enregistrement de ressources. Champ OBLIGATOIRE. Le
;    type sera explicite dans chaque exemple ;
;
;    - donnees : donnees associes au type d'enregistrement ;
;
; Cf. named(8).
;
; Debut 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 mel "root@machine1.domaine1".
;
;

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

;
; Les options qui suivent n'ont pas a etre modifiees 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 accociee au nom
; "localhost".
;

1    IN    PTR    localhost.
```

Voici le fichier `/var/named/domaine1` :

```
;
; Fichier /var/named/domaine1 - Contient les informations sur
; la zone domaine1 pour resoudre un nom d'hote en adresse IP.
;
; Ce fichier est appelle par le fichier /etc/named.conf. Le domaine
; actuel est celui specifie par la ligne appelant ce fichier :
;
;    zone "domaine1" {
;        type master;
;        file "domaine1";
;    };
```

```
;
; Le domaine local, represente par le caractere ‘‘@’’, 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 mel : tout mel doit etre envoye d’abord a machine1, puis a
; machine1.domaine1. Les nombres indiquent la priorite.
;

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

;
; Informations concernant chaque machine du reseau devant etre resolue
; par le serveur de noms local.
;
```

```
; Adresse de reference de machine1 : 192.168.1.1 (type A, unique pour une
; adresse). Diverses informations sur la machine peuvent etre specifiees
; 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"

;
; nougat, une autre machine.
;

nougat    IN    A        192.168.1.2
; nougat    IN    HINFO    "Non definit" "Non definit"

;
; localhost, pour l'interface de bouclage.
;

localhost   IN    A        127.0.0.1

;
; Quelque alias (CNAME) : un nom d'hote precise 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
; designent la meme 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/domaine1.rev` :

```
;
```

```
; Fichier /var/named/domaine1.rev - Contient les informations sur
; la zone domaine1 pour resoudre une adresse IP en nom d'hote.
;
; Ce fichier est appelle par le fichier /etc/named.conf. Le domaine
; actuel est celui specifie par la ligne appelant ce fichier :
;
;    zone "1.168.192.in-addr.arpa" {
;        type master;
;        file "domaine1";
;    };
;
; 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 donnees par rapport
; au domaine local. Ainsi, 2 <-> 2.1.168.192.in-addr.arpa.
;

1    IN    PTR    machine1.domaine1.
2    IN    PTR    nougat.domaine1.
```

1.5.2 Tester le nouveau réseau

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

```
$ 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

> machine3.domaine1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine3.domaine1
Address:  192.168.1.3

> 192.168.1.1
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine1.domaine1
Address:  192.168.1.1

> 192.168.1.2
Server:  machine1.domaine1
Address:  192.168.1.1

Name:    machine2.domaine1
Address:  192.168.1.2

> 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

> ls -d domaine1
[machine1.domaine1]
$ORIGIN domaine1.
@                       12H IN SOA      machine1 root.machine1 (
                                        2000070306      ; serial
                                        1H              ; refresh
                                        15M             ; retry
                                        2W              ; expiry
                                        12H )           ; minimum

                        12H IN NS       machine1
                        12H IN MX       10 machine1
                        12H IN MX       20 machine1
                        12H IN TXT      "Serveur DNS local de domaine1"
                        12H IN A        192.168.1.1
                        12H IN A        192.168.1.3
localhost               12H IN A        127.0.0.1
mail                    12H IN CNAME    machine1
www                     12H IN CNAME    machine1
                        12H IN A        192.168.1.1
                        12H IN A        192.168.1.2
ftp                     12H IN CNAME    machine1
@                       12H IN SOA      machine1 root.machine1 (
                                        2000070306      ; serial
                                        1H              ; refresh
                                        15M             ; retry
                                        2W              ; expiry
                                        12H )           ; minimum

>
```

1.6 Un 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)
 *
 * Specifie le(s) chemin(s) ou se trouvent les fichiers de zone DNS.
 *
 * Valeurs possibles : directory chemin
 */

options {
    directory "/var/named";

/*
 * Desactive 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 resoudre 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 resolution inverse. Les adresses IP commencant
 * par 127.0.0 peuvent etre resolues en nom d'hote dans le fichier specifie
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};

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

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

/*
 * 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 slave;
    file "domaine1.rev";
    masters { 192.168.1.1; };
};
```

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

La mise à jour du DNS “escalve” 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).

1.7 Configurer un 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 nouveau fichier `/etc/named.conf` :

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
 *
 * Specifie le(s) chemin(s) ou se trouvent les fichiers de zone DNS.
 *
 * Valeurs possibles : directory chemin
 */
```

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

/*
 * On est derriere un firewall
 */

    query-source address * port 53;

/*
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de resoudre 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 resolution inverse. Les adresses IP commencant
 * par 127.0.0 peuvent etre resolues en nom d'hote dans le fichier specifie
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};

/*
 * Fichier utilise pour la resolution des noms d'hote se terminant par
```

```
 * domaine1
 */

zone "domaine1" {
    type master;
    file "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 "domaine1.rev";
};
```

1.8 Configurer plusieurs domaines

On peut configurer plusieurs domaines pour un même DNS. Cela peut être interessant 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). Ces nouvaux fichiers auront la même syntaxe que dans la section 1.5.

1.8.1 Fichiers à écrire

```
/*
 * Fichier /etc/named.conf - Premier fichier lu par named. Cf. named.conf(5)
 *
 * Specifie le(s) chemin(s) ou se trouvent les fichiers de zone DNS.
 *
 * Valeurs possibles : directory chemin
 */

options {
    directory "/var/named";

/*
```

```
 * Adresse des serveurs a contacter si le serveur de noms local est incapable
 * de resoudre 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 resolution inverse. Les adresses IP commencant
 * par 127.0.0 peuvent etre resolues en nom d'hote dans le fichier specifie
 */

zone "0.0.127.in-addr.arpa" {
    type master;
    file "named.local";
};

/*
 * Fichier utilise pour la resolution des noms d'hote se terminant par
 * domaine1
 */

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

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

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

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

/*
 * Fichier utilise pour la resolution inverse. Les adresses IP commencant
 * par 192.168.2 peuvent etre resolues en nom d'hote dans le fichier
 * specifie
 */

zone "2.168.192.in-addr.arpa" {
    type master;
    file "domaine2.rev";
};

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

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

/*
```

```
 * Fichier utilise pour la resolution inverse. Les adresses IP commencant
 * par 192.168.3 peuvent etre resolues en nom d'hote dans le fichier
 * specifie
 */

zone "3.168.192.in-addr.arpa" {
    type master;
    file "domaine3.rev";
};
```

1.8.2 Tester notre nouvelle configuration

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

```
$ 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
> ls -d domaine2
[machine1.domaine1]
$ORIGIN domaine2.
@                       12H IN SOA      machine1.domaine1. root.machine1.domaine1. (
                                        2000070306      ; serial
                                        1H              ; refresh
                                        15M             ; retry
                                        2W              ; expiry
                                        12H )           ; minimum

                        12H IN NS       machine1.domaine1.
                        12H IN MX       20 machine1.domaine1.
                        12H IN TXT      "Serveur DNS local de domaine2"
                        12H IN A        192.168.1.1
                        12H IN A        192.168.2.3
localhost               12H IN A        127.0.0.1
mail                    12H IN CNAME    machine1
www                     12H IN CNAME    machine1
                        12H IN A        192.168.2.1
                        12H IN A        192.168.2.2
ftp                     12H IN CNAME    machine1
@                       12H IN SOA      machine1.domaine1. root.machine1.domaine1. (
                                        2000070306      ; serial
                                        1H              ; refresh
                                        15M             ; retry
                                        2W              ; expiry
                                        12H )           ; minimum

>
```

1.9 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 :

```
;
; Delegation 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 `domaine1.fr.rev` comme suit :

```
;
; Delegation 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.
```

1.10 Débugger un 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` :

```
# 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

>
```

1.11 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 utilise pour la resolution des noms d'hote se terminant par
 * domaine1. Seul le DNS d'adresse 192.168.2.1 a le droit de recuperer les
 * informations a partir de ce DNS.
 */

zone "domaine1" {
    type master;
    file "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 resolution des noms d'hote se terminant par
 * domaine1. Seul le DNS d'adresse 192.168.2.1 a le droit de recuperer les
 * informations a partir de ce DNS. Seuls les hotes du domaine 192.168.1.0/24
 * sont autorises 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 :

```
# useradd -M -r -d /var/named -s /bin/false named
# 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 *rajouter SuSE*) pour y mettre la ligne :

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

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

Références

[1] **DNS HOW-TO**, Nicolai Langfeldt, 11 février 1999.
ftp://ftp.lip6.fr/pub/linux/french/docs/HOWTO/DNS-HOWTO.gz

[2] **Securing and Optimizing Linux: Red Hat Edition**, Gerhard Mourani, Août 2000.
http://www.linuxdoc.org/LDP/gawlso/Securing-Optimizing-Linux-RH-Edition-1_3.pdf
http://www.linuxdoc.org/LDP/gawlso/floppy.tgz

[3] **Administration Système UNIX**, Thierry BESANCON, Pierre DAVID, Joël MARCHAND, 1 avril 1996.
ftp://ftp.uvsq.fr/pub/sysadmin/cookbook/cookbook/2.0/acc.ps.gz

[4] **L'intégration du réseau de Jussieu dans le réseau Internet**, Pierre DAVID, Jacky THIBAULT, 22 avril 1991.
ftp://ftp.jussieu.fr/jussieu/doc/local/integ.ps.Z

[5] **Domain Name System - Fonctionnement et Installation**, Pierre DAVID, Jacky THIBAULT, 1 avril 1993.
ftp://ftp.jussieu.fr/jussieu/doc/local/dnsmail.ps.Z