Leçon 9 : Le service DNS

« Celui qui contrôle le DNS, contrôle le monde »

9.1 Introduction

Le service DNS est un des services les plus critiques et les plus utilisés sur Internet. En effet, c'est grâce à ce service qu'il est possible de *traduire un nom* en *une adresse IP*. Sans ce service, nous serions obligés de mémoriser des centaines d'adresses IP, ce qui, avouons-le, ne serait guère pratique (ni en IPv4 et encore moins en IPv6).

Ce service est également impliqué dans la livraison du courrier électronique en répondant à la question suivante : *« A qui adresser les mails à destination du nom de domaine @helmo.be ? »*. La réponse à cette question est donnée dans un enregistrement particulier du DNS, indiquant le serveur à contacter pour les mails.

Comme nous le voyons dans cette introduction, le DNS est important au bon fonctionnement d'Internet.

9.2 Rappels sur le fonctionnement du DNS

Comme annoncé dans l'introduction, le service DNS est responsable de *la traduction* d'un nom DNS en adresse IP (et inversement, d'une adresse IP vers un nom DNS). Il faut aussi rappeler qu'il s'agit d'un système hiérarchique dans lequel un domaine est responsable des sous-domaines qu'il contient. Ainsi, `helmo.be` est responsable des sous-domaines `student.helmo.be`, `cg.helmo.be` ou `co.helmo.be`.

Un serveur DNS peut *déléguer* la gestion d'un sous-domaine à un autre serveur. Ainsi, le serveur DNS `.be` a délégué la gestion de `helmo.be` au serveur `193.190.64.113`. Pour des raisons de sécurité et de performance, il est très courant de disposer de *serveurs DNS secondaires* complètement et automatiquement synchronisé avec le serveur DNS principal.

9.2.1 Un peu de vocabulaire

Un **domaine** est un nom DNS complètement qualifié, qui peut être *local* (interne à une entreprise) ou *officiel* (acheté auprès d'un registrar). Ainsi, par convention, les *domaines locaux* portent le suffixe `.local` ou `.localdomain`.

Un **serveur DNS** est une machine exécutant le service `bind`[27] qui permet de répondre aux requêtes DNS.

Une **zone DNS** reprend *la liste de tous les enregistrements* d'un domaine. Elle peut être *directe* (transformation d'un nom en adresse IP) ou *inverse* (transformation d'une adresse IP en nom). Ainsi, la zone DNS directe attachée au domaine `helmo.be` contient des enregistrements pour les noms `www.helmo.be`, `elearning.helmo.be` ou encore `webmail.helmo.be`.

[27] Par exemple – il existe d'autres serveurs DNS mais `bind` est celui distribué en standard dans CentOS 7

Un **enregistrement** est une entrée dans une zone DNS associée à un domaine. On trouve des enregistrements de plusieurs types :

Type d'enregistrement	Explication
A	Permet de renseigner l'adresse IPv4 associée à un nom
AAAA	Permet de renseigner l'adresse IPv6 associée à un nom
CNAME	Permet de faire pointer un nom vers un enregistrement de type A, AAAA ou PTR. Il n'est pas autoriser de faire pointer un CNAME vers un autre CNAME.
MX	Permet de renseigner les noms des serveurs mails à contacter pour le domaine ou le sous-domaine
NS	Permet de renseigner les noms des serveurs DNS en charge du domaine ou du sous-domaine mentionné
PTR	Permet de renseigner le nom associé à l'adresse IP considérée. Uniquement dans les zones DNS inverses.

9.2.2 DNS à HELMo

La haute école HELMo dispose de plusieurs serveurs DNS. Ainsi, il y a :

```
$ host -t NS helmo.be
helmo.be name server ns.helmo.be.
helmo.be name server ns2.helmo.be.
helmo.be name server ns6.gandi.net.
```

La commande `host` (que nous détaillerons plus loin) permet d'interroger le service DNS. Ainsi, nous lui demandons ici les enregistrements de type NS pour le domaine `helmo.be`. Nous obtenons 3 serveurs responsables de la zone helmo.be : le serveur principal `ns.helmo.be` et les 2 serveurs secondaires `ns2.helmo.be` et `ns6.gandi.net`.

Nous pouvons également savoir les serveurs qui gèrent le mail :

```
$ host -t MX helmo.be
helmo.be mail is handled by 10 smtp.helmo.be.
helmo.be mail is handled by 20 smtp2.helmo.be.
```

*Nous apprenons ici qu'il y a 2 serveurs gérant le courrier à HELMo : le serveur principal `smtp.helmo.be` et le serveur secondaire `smtp2.helmo.be`. Les valeurs `10` et `20` représentent **la priorité** avec laquelle le serveur doit être contacté : plus ce nombre est faible, plus ce serveur est prioritaire. Ainsi, c'est d'abord le serveur `smtp.helmo.be` qui doit être contacté. Si celui-ci ne peut répondre, le serveur de backup `smtp2.helmo.be` peut alors être contacté.*

9.3 DNS interne et DNS externe

Il arrive très souvent que la réponse du serveur DNS doive être **adaptée en fonction de l'origine de la requête**. Ainsi, si l'on se trouve dans le réseau interne, il est souvent nécessaire que le serveur DNS retourne l'adresse IP interne correspondant à la requête. Par contre, si la requête provient de l'extérieur du réseau, il faut alors que la réponse mentionne l'adresse IP publique du serveur.

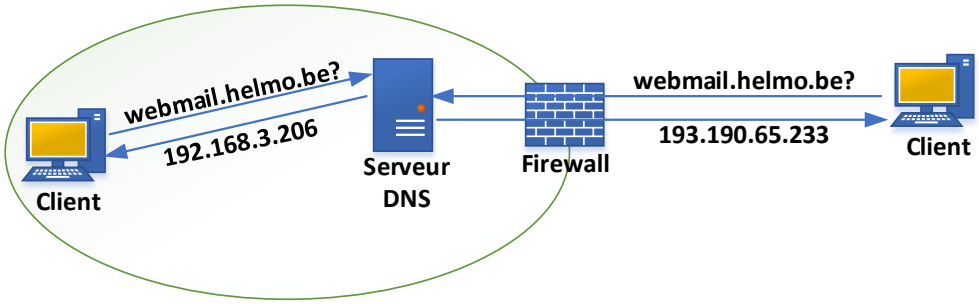

Figure 9.1 : réponse différente en fonction de l'origine

Ainsi, comme le montre la figure 9.1, la réponse pour la requête `webmail.helmo.be` sera différente selon que le client se trouve à l'intérieur du réseau de HELMo (et donc l'adresse IP interne `192.168.3.206` est retournée) ou à l'extérieur de celui-ci (et, dans ce cas, c'est l'adresse IP publique `193.190.65.233` qui est retournée).

9.4 Configuration du service DNS

Si nécessaire, les packages DNS peuvent être installés :

```
$ yum install bind-chroot
```

La version *chroot* est une installation sécurisée du service DNS. C'est la raison pour laquelle c'est cette version qui est installée et utilisée.

La configuration est concentrée en plusieurs endroits :

- Le fichier `/var/named/chroot/etc/named.conf` – Ce fichier doit lister les différents domaines configurés et gérés par le serveur DNS. Dans ce fichier, chaque domaine est lié au fichier décrivant les enregistrements de zone DNS. Par exemple, celui de HELMo :
  - Liste tous les domaines que le serveur DNS gère. On y trouve ainsi les domaines `helmo.be`, `helmo.eu`, `student.helmo.be`, `cg.helmo.be` ou encore `co.helmo.be`.
  - Pour chaque domaine listé, un fichier de zone, placé dans le dossier `/var/named/chroot/var/named` est mentionné.
- Le dossier `/var/named/chroot/var/named` – Ce dossier doit contenir les fichiers décrivant les zones DNS. Ainsi, il convient de créer un fichier par nom DNS souhaité (au moins) et d'y mentionner, à l'intérieur, les enregistrements de zone. Ainsi, à HELMo,
  - Le fichier décrivant les enregistrements de la zone `helmo.be` se trouve dans `/var/named/chroot/var/named/db.helmo.be`. A l'intérieur, on trouve les enregistrements `www`, `elearning`, `webmail`, … et les adresses IP correspondantes.

Dans la suite, nous allons détailler pas à pas la configuration d'un nouveau serveur DNS.

9.4.1 Etape 1 : récupérer le modèle de configuration

Dans cette étape, nous allons récupérer le modèle de fichier de configuration. Une fois ce modèle récupéré, nous pourrons modifier celui-ci.

```
$ cd /var/named/chroot/etc
$ cp /usr/share/doc/bind-9.9.4/sample/etc/* .
```

Cette dernière commande copie les fichiers `named.conf` et `named.rfc1912.conf` dans le dossier `/var/named/chroot/etc`.

9.4.2 Etape 2 : configuration du fichier named.conf

Nous allons, dans cette étape, parcourir le fichier `named.conf` et pointer les éléments de configuration importants.

Section options

Dans la section `options`, il est possible de définir les *options globales* applicables sur le serveur DNS. Le paramètre `directory` désigne le répertoire de travail. Les options `dump-file`, `statistics-file` et `memstatistics-file` sont bien configurées par défaut.

Voici quelques options importantes qu'il peut être nécessaire d'ajouter/d'adapter :

Option	Explication
listen-on	Détermine sur quelles adresses IPv4 le serveur DNS écoute. Par défaut, il s'agit uniquement `127.0.0.1`, ce qui posera un problème. Il est possible de changer cette option en `any`. Par défaut : `listen-on port 53 { 127.0.0.1; };`
listen-on-v6	Détermine sur quelles adresses IPv6 le serveur DNS écoute. Par défaut, il s'agit uniquement `::1`, ce qui posera un problème. Il est possible de changer cette option en `any`. Par défaut : `listen-on-v6 port 53 { ::1; };`
allow-query	Détermine à partir d'où les requêtes sont autorisées. **allow-query** concerne les domaines gérés par le serveur DNS. Par défaut, seul `localhost` est accepté comme origine. Il est préférable de placer cette option à `any` pour autoriser le serveur à répondre sur les domaines qu'il gère. Par défaut : `allow-query { localhost; } ;`
allow-query-cache	Détermine à partir d'où les requêtes sont acceptées. **allow-query-cache** concerne les autres requêtes (celles qui ne concernent pas les domaines gérés par le serveur). Ainsi, le serveur doit-il répondre à « Qui est www.google.com ? » à ce client ? Par défaut, seul `localhost` est accepté comme origine. Cette option devra également être modifiée pour répondre aux requêtes extérieures. Par défaut : `allow-query-cache { localhost; };`
version	Cette option détermine la chaîne qui sera retournée si un client demande la version du serveur *bind*. Il est intéressant de définir une valeur pour des raisons de sécurité. Ex : `version "1234";`
recursion	Cette option détermine si le serveur interroge les autres serveurs pour obtenir une réponse (est-il configuré pour donner une réponse au client ou bien se limite-t-il à retourner la liste des serveurs mondiaux s'il ne gère pas le nom DNS demandé. Par défaut : `recursion yes;`
dnssec-enable **dnssec-validation** **dnssec-lookaside**	Ces options activent les extensions de sécurité dnssec. Si DNSSEC n'est pas utilisé, il est possible de désactiver celui-ci. Par défaut : `dnssec-enable yes;`

	`dnssec-validation yes ;` `dnssec-lookaside auto;`
forwarders	Cette option liste les serveurs DNS vers lesquels ce serveur peut faire suivre les requêtes pour les domaines qu'il ne gère pas. Cette option est utilisée pour alléger le serveur DNS en transmettant les requêtes à un autre serveur qui est alors chargé de faire la résolution. Cette option est également nécessaire si l'administrateur réseau a limité les serveurs DNS autorisés. Par exemple : `forwarders { 192.168.128.2; };`

Les vues

Le fichier de configuration comporte plusieurs sections `view`. Ces sections **permettent d'adapter la réponse** du serveur **en fonction de l'origine de la requête**. Ainsi, si la requête provient du réseau interne, il faut s'assurer que la réponse apportée sera l'adresse IP interne alors que si la requête provient de l'extérieur, l'adresse IP publique doit être retournée (voir figure 9.1).

La structure de la description d'une vue est :

```
view "nom_de_la_vue"
{
  match-clients   { ... } ;
  recursion ... ;
  ...
}
```

Les vues sont directement liées à l'option `match-clients` qui impose que la vue choisie est sélectionnée si l'origine de la requête est listée dans `match-clients`. Il est possible de lister des adresses IP, des préfixes IP sous la forme ADRESSE/MASQUE, ou encore d'utiliser les mots clés `localhost` (cette machine seulement), `localnets` (tous les réseaux auxquels cette machine est directement connectée) ou `any` (n'importe quelle origine).

Exemple :

```
match-clients { 192.168.5.0/24; 192.168.3.5; 2001:6a8:2cc0:8020::/64; };
match-clients { localnets; };
match-clients { any; };
```

Le modèle de configuration propose les 3 vues suivantes dans cet ordre (en partant du plus spécifique au plus général) :

Nom de la vue	match-clients	Explication
localhost_resolver	`localhost`	Cette vue est utilisée par le serveur lorsque la requête émane de lui-même. Il est important d'y lister toutes les zones connues du serveur et de mentionner l'option `recursion yes;`
internal	`localnets`	Cette vue est utilisée par le serveur lorsque la requête émane d'un réseau auquel il est directement connecté (réseau interne donc). Il est important d'y lister toutes les zones connues du serveur et d'y ajouter l'option `recursion yes;`
external	`any`	Cette vue est utilisée par le serveur lorsque la requête

		provient d'ailleurs (ni du serveur lui-même, ni d'un réseau directement relié à ce serveur). Il est important de lister uniquement les zones gérées par ce serveur et de préciser l'option `recursion no;` afin que le serveur ne réponde pas aux requêtes pour lequel il ne fait pas autorité (ie. Les domaines qu'il ne gère pas).

La description d'une zone directe

A **l'intérieur de chaque vue**, nous trouvons la liste des domaines gérés par ce serveur. Il est possible de lister ces domaines dans le fichier `named.conf` ou, via l'option `include`, de les spécifier dans un fichier annexe.

Comme le nombre de zone ne sera pas trop important ici, nous choisirons de les lister dans le fichier `named.conf`.

La description d'une zone prend toujours la forme suivante :

```
zone "nom.dns.de.la.zone" {
  type master|hint;
  file "fichierDeZone.db";
  allow-update { none; };
  allow-transfer { none; };
};
```

On commence par donner la zone DNS en question. Ainsi pour les serveurs de HELMo, nous avons une entrée zone "helmo.be". Ensuite, on doit mentionner le type de zone : `hint` (uniquement valide pour la zone « . » qui mentionne les serveurs DNS mondiaux) ou `master` (indique que le serveur gère la zone DNS en question). Donc pour une zone directe (sur le serveur principal, nous aborderons le serveur de backup plus loin), ce sera toujours master.

Ensuite, avec `file`, il faut donner le fichier contenant les enregistrements DNS de cette zone. Dans le fichier mentionné, nous trouverons tous les noms des machines et les adresses IP associées à ces noms. Il s'agit du fichier de zone correspondant dont nous détaillerons le contenu à l'étape suivante.

L'option `allow-update` est utilisée dans la configuration des zones DNS dynamiques (par exemple lorsqu'un serveur DHCP ajoute dynamiquement les entrées quand un nouveau client se connecte). Nous n'étudierons pas cette possibilité, raison pour laquelle les mises à jour dynamiques sont refusées.

Enfin, l'option `allow-transfer` détermine vers quels serveurs DNS le transfert de zone est autorisé. Il s'agit d'une option à prendre en compte lorsqu'on dispose d'un serveur principal et d'un ou plusieurs serveurs secondaires (ou backups) qui doivent se synchroniser avec le serveur principal.

Par exemple, à HELMo, la description de la zone `helmo.be` dans la vue `internal` prend la forme :

```
zone "helmo.be" in {
        type master;
        file "db.helmo.be.internal[28]";
```

[28] Afin de faire la distinction entre les réponses internes (avec les adresses IP privée) et externes (avec les adresses IP publiques), **des fichiers de zones différents sont utilisés**. Ainsi, dans la vue interne, il y a un fichier

```
                allow-update { none; };
                allow-transfer { 192.168.3.30; 192.168.3.19; 192.168.3.208;
192.168.0.2; 10.0.0.2; 192.168.0.10; 192.168.224.200; 192.168.20.2;
192.168.20.3; 192.168.3.206; 192.168.27.2; 192.168.128.2; };
        };
```

Comme nous pouvons le voir dans le fichier named.conf, celui-ci liste des zones en exemple comme *my.internal.zone, my.slave.zone, my.ddns.internal.zone* et *my.external.zone*. **Toutes ces zones doivent être supprimées ou adaptées en fonction des besoins**.

La description d'une zone inverse

Décrire une zone inverse est semblable à la définition d'une zone directe. Pour rappel, la zone inverse est utilisée lorsqu'il faut *traduire* une adresse IP en nom. A l'exception du nom de la zone, qui est très particulier, on y trouve les mêmes informations que pour la zone directe. Ainsi, le nom de la zone est formé, le plus souvent, par les 3 derniers octets de l'adresse IP, inversé et suffixé par `in-addr.arpa`.

Nom de la zone	Utilisée pour traduire les IP
`128.168.192.in-addr.arpa`	`192.168.128.x` ➔ nom DNS
`1.0.10.in-addr.arpa`	`10.0.1.x` ➔ nom DNS

Exemple de définition d'une zone inverse :

```
        zone "21.168.192.in-addr.arpa" in {
                type master;
                file "db.192.168.21";
                allow-update { none; };
                allow-transfer { 192.168.3.30; 192.168.3.19; 192.168.3.208;
192.168.0.2; 10.0.0.2; 192.168.0.10; 192.168.224.200; 192.168.20.2;
192.168.20.3; 192.168.3.206; 192.168.27.2; 192.168.128.2; };
        };
```

Comme nous pouvons le voir, celle-ci est très proche de la définition d'une zone directe.

La description d'une zone de backup (ou slave, secondaire)

La description d'une zone de backup (également appelée *slave* ou *secondaire*) ne demande pas beaucoup de configuration. On définit le nom de la zone (identique à celle définie sur le serveur maître), on place le type à `slave`, on mentionne le fichier décrivant cette zone et on mentionne, avec l'option `masters`, le ou les serveurs DNS maîtres.

Sur les serveurs DNS maîtres, il **faut autoriser le transfert de zone**. Pour ce faire, il faut que l'IP du serveur de backup apparaisse dans la section `allow-transfer` du serveur maître.

Par exemple, sur le serveur DNS secondaire de HELMo (`ns2.helmo.be`), nous avons la déclaration suivante pour le transfert de la zone `helmo.be` :

---

de zone `db.helmo.be.internal` (listant des IP privées) alors que la vue externe référence le fichier `db.helmo.be.external` (listant des IP publiques).

```
zone "helmo.be" in {
        type slave;
        file "slaves/db.helmo.be.internal";
        allow-transfer { none; };
        masters { 192.168.3.209; };
};
```

9.4.3 Etape 3 : Les enregistrements d'une zone DNS

Nous avons jusqu'à présent indiqué au serveur DNS le nom des domaines qu'il devait gérer (par exemple `mondomaine.be` ou `monautredomaine.org`). Pour chacun d'eux, nous avons mentionné un fichier de zone. Nous allons maintenant détailler les enregistrements DNS qu'il est possible d'indiquer dans ces fichiers de zone pour permettre au serveur de répondre à des requêtes comme « qui est `www.mondomaine.be` ? » ou « `webmail.monautredomaine.org` ? ».

Les fichiers décrivant les enregistrements des zones doivent être placés dans le dossier `/var/named/chroot/var/named/`.

Récupération des modèles de fichiers de zone

Nous allons commencer par récupérer les modèles des fichiers de zone :

```
$ cd /var/named/chroot/var/named/
$ cp -a /usr/share/doc/bind-9.9.4/sample/var/named/* .
$ chown -R named.named *
$ rm my.* slaves/my.*
```

Nous avons récupéré les dossiers `data` et `slaves` ainsi que les fichiers `named.ca` (reprenant la liste des serveurs DNS mondiaux, renseigné par la zone « . »), `named.empty`, `named.localhost` et `named.loopback`. Grâce à la commande `chown`, nous changeons le propriétaire de tous les fichiers et dossiers par l'utilisateur `named` et le groupe `named`. Enfin, la commande `rm` supprime les zones parasites et vides données à titre d'exemple.

Ajout du fichier de zone directe

Le fichier qui doit contenir les enregistrements DNS doit porter le nom référencé par l'option `file` de la zone correspondante. Dans l'exemple de HELMo donné ci-avant, le nom du fichier doit être `db.helmo.be.internal`.

Ce fichier texte doit contenir des éléments particuliers. Ainsi l'en-tête commence ainsi :

```
$TTL  86400
@     IN    SOA   monserver.mondomaine.be.      admin.mondomaine.be. (
                                2015080801  ; Serial
                                28800       ; Refresh
                                14400       ; Retry
                                3600000     ; Expire
                                3600 )      ; Name Error
```

L'entête comment par `$TTL` qui indique, en seconde, la durée pendant laquelle les informations peuvent être mise en cache. Ainsi, une valeur de 0 mentionnerait que ces données ne peuvent pas être mise en cache.

La seconde ligne contient les éléments suivants, **tous séparés par des tabulations** :

Elément	Explication
@	Référence automatiquement le nom de la zone actuelle. Nous aurions pu indiquer mondomaine.be.
IN SOA	Indique quel serveur fait autorité sur la zone. Le nom du serveur, complètement qualifié (et donc se terminant par un `.`) est mentionné ensuite.
monserver.mondomaine.be.	Nom du serveur principal faisant autorité pour la zone définie (ici `mondomaine.be.`). Le nom du serveur doit être *complètement qualifié.*
admin.mondomaine.be.	Adresse mail du gestionnaire de la zone (pour laquelle le symbole « `@` » a été changé par « `.` »)
2015080801	Numéro de série attaché à la zone DNS. A chaque changement, il faut mettre à jour (incrémenter par exemple) celui-ci. Ce numéro de série permet aux serveurs secondaires de se synchroniser correctement. Traditionnellement, le numéro de série est écrit sous la forme `AAAAMMJJXX` (avec `AAAAMMJJ` la date de création du fichier de zone et `XX` la version actuelle).
28800	Indique le temps, en secondes, après lequel les serveurs backups (ou secondaires) se resynchroniseront (ici 8 heures).
14400	Indique le temps, en seconde, entre les tentatives de connexion aux serveurs maîtres lorsque celui-ci ne répond plus (ici 4 heures).
3600000	Indique le temps, en secondes, après lequel les serveurs secondaires considérerons que l'information est périmée quand ils n'arrivent plus à contacter les serveurs principaux (ici 1000 heures, soit 41 jours).
3600	Indique le temps, en secondes, pendant lequel un réponse de type NXDOMAIN peut-être conservée par un serveur (ici 1 heure). En cas de défaillance du DNS ou de nom inexistant, les serveurs qui tentent de résoudre un nom sur mondomaine.be généreront une erreur de type NXDOMAIN. Cette réponse peut être mise en cache par le serveur pendant 1 heure maximum (après, il doit à nouveau tenter de résoudre le nom, si celui-ci est demandé à nouveau).

Une grande difficulté dans la gestion du DNS sur Internet est la présence *des caches* sur tous les serveurs. En effet, pour éviter d'interroger plusieurs fois le DNS pour les mêmes noms, ceux-ci sont maintenus en caches par les serveurs DNS. Cela améliore les performances et diminue le trafic DNS global. Cependant, le désavantage est qu'en cas de modification de la zone DNS, la propagation de cette modification **peut prendre jusqu'à 24h**.

Après cette entête, nous trouverons les enregistrements de type `NS`, `MX`, `A`, `AAAA` et `CNAME`. A titre d'exemple illustratif, voici un extrait du fichier de zone `db.helmo.be.internal` :

```
$TTL    86400
@       IN      SOA     ns.helmo.be.    hostmaster.helmo.be. (
                                2008011117      ; Serial
                                28800           ; Refresh
                                14400           ; Retry
```

```
                                3600000         ; Expire
                                3600 )          ; Name error


; CRITICAL INFORMATION
; ** Name server
                        IN      NS      ns.helmo.be.
                        IN      NS      ns2.helmo.be.
                        IN      NS      ns6.gandi.net.
; ** Main mail server
helmo.be.               IN      MX      10      smtp.helmo.be.
                        IN      MX      20      smtp2.helmo.be.
student.helmo.be.       IN      MX      10      smtp.helmo.be.
                        IN      MX      20      smtp2.helmo.be.

; COMPUTER
helmo.be.               IN      A       192.168.3.203
ns1                     IN      A       192.168.3.206
                        IN      AAAA    2001:6a8:2cc0:8000::206
webmail                 IN      CNAME   ns1.helmo.be.
www                     IN      A       192.168.3.203
```


Dans l'extrait présenté, juste après l'entête, nous avons l'enregistrement `NS` avec la liste de tous les serveurs DNS (principal et secondaires) gérant cette zone. Cet un élément important car ces informations sont utilisées pour notifier aux serveurs les modification de zones.

Nous avons ensuite, avec `MX`, la liste des serveurs mails utilisés et leurs priorités.

Enfin, nous avons les différentes entrées créées dans la zone (ici `ns1.helmo.be`, `webmail.helmo.be` et `www.helmo.be`). Il faut noter qu'un nom qui n'est pas complètement qualifié (et qui ne se termine pas par un « . ») se verra automatiquement suffixer par le nom de la zone.

Ainsi `ns1` désigne `ns1.helmo.be`. Nous aurions pu écrire, également `ns1.helmo.be.` qui est complètement qualifié.

Enfin, rappelons que `A` permet d'associer le nom et l'adresse IPv4 alors que `AAAA` permet d'associer le nom et l'adresse IPv6 et que `CNAME` représente un raccourci.

Ajout d'un fichier de zone inverse

Pour les zones DNS inverses, la configuration est semblable à celle que nous venons de voir. Ainsi, nous retrouvons une entête similaire à celle de la zone directe et puis les enregistrements `NS` et enfin nous avons, pour les adresses IP, le nom vers lequel elle pointe (enregistrements `PTR`).

Voici un exemple de fichier de zone inverse pour 192.168.21.x :

```
$TTL    86400
@       IN      SOA     ns.helmo.be.    hostmaster.helmo.be. (
                                2013072703      ; Serial
                                28800           ; Refresh
                                14400           ; Retry
                                3600000         ; Expire
```

```
                                3600 )          ; Name error

; CRITICAL INFORMATION
; ** Name server
            IN      NS      ns.helmo.be.
            IN      NS      ns2.helmo.be.

; PTR RECORD
1           IN      PTR     sw-hp1910-campusguillemins.net.helmo.be.
2           IN      PTR     sw-hp2920-campusourthe.net.helmo.be.
252         IN      PTR     cp4600-stmartin.net.helmo.be.
253         IN      PTR     sw-hp2920-campusguillemins.net.helmo.be.
```

Il faut ainsi comprendre que l'adresse IP `192.168.21.1` pointe vers le nom `sw-hp1910-campusguillemins.net.helmo.be`. Il en va de même pour les autres entrées.

9.4.4 Démarrer le service DNS

Pour démarrer le service DNS, il faut utiliser la commande `systemctl` comme suit :

```
$ systemctl start named-chroot
```

Pour activer le service au démarrage de la machine :

```
$ systemctl enable named-chroot
```

9.4.5 Configurer les postes clients

Configurer les postes clients signifie que ces derniers utiliseront le service DNS installé sur cette machine. Il faut donc modifier les paramètres DNS utilisés. Cela peut se faire de plusieurs manières :

- Si un serveur DHCP est utilisé pour configurer automatiquement les postes clients, il est nécessaire de modifier la configuration de ce service pour qu'il renseigne l'adresse IP du nouveau serveur DNS.
- Si la configuration IP est statique sur les postes clients, il **faut se référer à la leçon décrivant la configuration réseau** pour adapter les serveurs DNS utilisés.

9.5 Les outils DNS

Il existe quelques outils permettant d'interroger, de manière directe, les serveurs DNS souhaités. Ces outils sont particulièrement intéressants pour déterminer si le serveur DNS répond correctement aux requêtes.

9.5.1 nslookup

L'outil nslookup est standard sur toutes les plateformes (Windows, Linux et MacOS X). Il permet d'interroger, de manière interactive, un serveur DNS.

```
$ nslookup
> www.helmo.be
Server:     192.168.3.209
```

```
Address:    192.168.3.209#53

Name: www.helmo.be
Address: 192.168.3.203
> 192.168.21.253
Server:           192.168.3.209
Address:    192.168.3.209#53

253.21.168.192.in-addr.arpa   name = sw-hp2920-campusguillemins.net.helmo.be.
> set type=MX
> helmo.be
Server:           192.168.3.209
Address:    192.168.3.209#53

helmo.be    mail exchanger = 20 smtp2.helmo.be.
helmo.be    mail exchanger = 10 smtp.helmo.be.
```

Comme nous pouvons le voir, il est possible d'obtenir des réponses pour des requêtes sur la zone directe, sur la zone inverse et même spécifier le type d'enregistrement souhaité.

9.5.2 host

La commande `host` permet le même type d'interrogation que `nslookup`. Elle n'est, cependant, pas interactive puisque l'utilisateur doit décrire sa requête par la ligne de commande.

```
$ host www.helmo.be
www.helmo.be has address 192.168.3.203

$ host -t NS helmo.be
helmo.be name server ns2.helmo.be.
helmo.be name server ns6.gandi.net.
helmo.be name server ns.helmo.be.

$ host 192.168.21.252
252.21.168.192.in-addr.arpa domain name pointer cp4600-stmartin.net.helmo.be.
```

9.5.3 dig

Le dernier outil intéressant pour interroger le service DNS est `dig`. A la différence des outils précédents, `dig` permet également de *tracer* une requête DNS, c'est à dire de visualiser tous les serveurs interrogés et les réponses apportées. Cet outil est particulièrement apprécié dans l'analyse des problèmes DNS.

`dig` permet d'effectuer beaucoup de tâches, je vous renvoie vers la page de manuel pour plus d'explication sur cet outil. Nous nous contenterons, ici, d'explorer la possibilité de *tracer* une requête.

```
$ dig +trace www.swila.be

; <<>> DiG 9.7.3-RedHat-9.7.3-1.el5 <<>> +trace www.swila.be
;; global options: +cmd
.                   449507      IN    NS    f.root-servers.net.
.                   449507      IN    NS    k.root-servers.net.
.                   449507      IN    NS    h.root-servers.net.
.                   449507      IN    NS    l.root-servers.net.
```

```
.                   449507      IN    NS    c.root-servers.net.
.                   449507      IN    NS    d.root-servers.net.
.                   449507      IN    NS    b.root-servers.net.
.                   449507      IN    NS    m.root-servers.net.
.                   449507      IN    NS    j.root-servers.net.
.                   449507      IN    NS    e.root-servers.net.
.                   449507      IN    NS    a.root-servers.net.
.                   449507      IN    NS    g.root-servers.net.
.                   449507      IN    NS    i.root-servers.net.
;; Received 272 bytes from 192.168.3.209#53(192.168.3.209) in 40 ms

be.                 172800      IN    NS    a.ns.dns.be.
be.                 172800      IN    NS    b.ns.dns.be.
be.                 172800      IN    NS    c.ns.dns.be.
be.                 172800      IN    NS    d.ns.dns.be.
be.                 172800      IN    NS    x.ns.dns.be.
be.                 172800      IN    NS    y.ns.dns.be.
;; Received 397 bytes from 2001:7fd::1#53(k.root-servers.net) in 11 ms

swila.be.           86400 IN    NS    ns1.sllabs.net.
swila.be.           86400 IN    NS    ns6.gandi.net.
swila.be.           86400 IN    NS    ns2.sllabs.net.
;; Received 100 bytes from 2001:dcd:7::8#53(y.ns.dns.be) in 8 ms

www.swila.be.           172800      IN    A     178.32.45.186
;; Received 46 bytes from 2001:41d0:2:25a8:ff10::10#53(ns1.sllabs.net) in
14 ms
```

La 1ère étape consiste à trouver les serveurs mondiaux (la zone « . »). Un de ces serveurs est interrogé (`k.root-servers.net`) et nous retourne la liste des serveurs gérant la zone « .be ». Un de ces serveurs est interrogé (`y.ns.dns.be`) et nous renvoie la liste des serveurs gérant la zone « swila.be ». A nouveau, un de ces serveurs est interrogé (`ns1.sllabs.net`) et nous renvoie la réponse demandée `178.32.45.186`.

9.6 Exercices

On vous demande de :

1) Configurer un serveur DNS pour la zone `<nomdefamille>.cg.helmo.be`.
   a) Dans la « vue locale », préciser les enregistrements suivants :
      - Entrée NS : `ns.<nomdefamille>.cg.helmo.be`
      - Les adresses suivantes :
        - `ns.<nomdefamille>.cg.helmo.be` ➔ `192.168.190.x` (IP de votre serveur)
        - `gate.<nomdefamille>.cg.helmo.be` ➔ IP de pfSense
        - `pfsense.<nomdefamille>.cg.helmo.be` ➔ CNAME vers `gate`

   b) Dans la « vue externe », préciser les enregistrements suivants :
      - Entrée NS : `gate.<nomdefamille>.cg.helmo.be`
      - Les adresses suivantes :
        - `ns.<nomdefamille>.cg.helmo.be` ➔ CNAME `gate`

- `gate.<nomdefamille>.cg.helmo.be` ➔ IP WAN de pfSense
- `pfsense.<nomdefamille>.cg.helmo.be` ➔ CNAME vers `gate`

c) Configurer votre serveur pour qu'il interroge le service DNS que vous venez de configurer. Activer ce service DNS au démarrage de la machine.

d) Modifier la configuration réseau de votre serveur pfSense pour ouvrir le port 53 et le rediriger vers votre serveur Linux. Votre serveur DNS est ainsi accessible à distance.

e) Tester votre configuration à partir du poste Windows. Vous pouvez utiliser l'outil `nslookup` pour valider votre configuration.

2) Configurer votre serveur DNS pour qu'il soit serveur secondaire de votre voisin. Pour ce faire, vous utiliserez l'adresse IP WAN.

3) Ajouter une zone DNS inverse pour les IPs `192.168.190.x`. Ajoutez-y la machine Windows, le firewall pfSense et votre serveur Linux.

« Celui qui contrôle le DNS, contrôle le monde »

Comme nous l'avons vu, l'administrateur du serveur DNS peut configurer celui-ci comme il veut. Rien n'empêche d'indiquer dans votre serveur que vous gérez la zone `facebook.com` ou `google.com`[29]. Ces manipulations vous permettraient ainsi de *capturer* le trafic à destination de ces sites sur votre réseau, d'afficher un avertissement, …

Une extension du DNS appelée DNSSEC permet de s'assurer que les informations obtenues auprès du serveur sont licites. Ainsi, pour www.swila.be, DNSSEC est activé et permet de vérifier **depuis les serveurs mondiaux jusqu'au domaine swila.be** que l'entrée www pointe bien vers l'adresse IP licite.

Nous remarquons cependant que DNSSEC se déploie timidement aujourd'hui. Cette leçon étant déjà assez longue et compliquée à appréhender sans la présentation de cette extension, j'ai choisi de ne pas l'aborder. Cependant, je ne peux que vous encourager à déployer ce mécanisme si vous devez mettre en place un serveur DNS sur Internet.

[29] A ce titre, Google a pris des mesures de sécurité importantes dans son navigateur Chrome pour éviter ce genre de manipulations. En effet, la clé publique des certificats utilisés par Google est connue du navigateur et celui-ci peut donc déterminer s'il est sur le site licite de Google ou non. De plus, le déploiement progressif de la technologie *certificate transparency* permet également de contrer ces manipulations pour les sites protégés par des certificats.