Stage d'été

Juillet 2009

Sujet : Serveur de nom domaine DNS

Propposé par : Bayrem JRIDI
Encadré par : M.Walid Boudiche

I- Introduction au DNS:

1) Définition:

Le **Domain Name System** (ou **DNS**, système de noms de domaine) est un service permettant d'établir une correspondance entre une adresse IP et un nom de domaine et, plus généralement, de trouver une information à partir d'un nom de domaine. À la demande de Jon Postel, Paul Mockapetris inventa le « *Domain Name system* » en 1983 et écrivit la première implémentation.

2) Associer une adresse IP et un nom de domaine:

Les ordinateurs connectés à un réseau IP, par exemple Internet, possèdent tous une adresse IP. Ces adresses sont numériques afin d'être plus facilement traitées par une machine. Selon IPv4, elles prennent la forme *xxx.yyy.zzz.aaa*, où xxx, yyy, zzz et aaa sont quatre nombres variant entre 0 et 255 (en système décimal). Selon IPv6, les IP sont de la forme aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh, où a, b, c, d, e, f, g et h représentent des caractères au format hexadécimal. Il n'est pas évident pour un humain de retenir ce numéro lorsque l'on désire accéder à un ordinateur d'Internet. C'est pourquoi un mécanisme a été mis en place pour permettre d'associer à une adresse IP un nom intelligible, humainement plus simple à retenir, appelé nom de domaine. Résoudre un nom de domaine, comme par exemple *fr.wikipedia.org*, c'est trouver l'adresse IP qui lui est associée.

a) Fichier *HOSTS*

Avant le DNS, la résolution devait se faire grâce à un fichier texte appelé HOSTS, local à chaque ordinateur. Sous UNIX, il se trouve dans le répertoire /etc. Sous Windows, il se trouve par défaut dans %SystemRoot%\system32\drivers\etc.

Dans ce fichier, chaque ligne correspond à une adresse IP à laquelle peuvent être associés un ou plusieurs noms de domaine. Ce système pose un problème de maintenance car le fichier doit être recopié sur tous les ordinateurs du réseau. On ne peut pas non plus organiser hiérarchiquement les domaines. C'est pour résoudre ce problème que Paul Mockapetris a mis au point le DNS en 1983.

b) Résolution et résolution inverse avec DNS

Avec DNS, la résolution se fait par l'intermédiaire d'un serveur. Quand un utilisateur souhaite accéder à un serveur web, par exemple celui de fr.wikipedia.org, son ordinateur émet une requête spéciale à un serveur DNS, demandant 'Quelle est l'adresse de *fr.wikipedia.org* ?'. Le serveur répond en retournant l'adresse IP du serveur, qui est dans ce cas-ci, 91.198.174.2.

Il est également possible de poser la question inverse, à savoir 'Quel est le nom de domaine ou quels sont les noms de domaines de telle adresse IP ?'. On parle alors de **résolution inverse** (inverse query) ou de **PTR Lookup** en référence à l'enregistrement DNS de type PTR.

Note : Cette déclaration inverse est importante sur les adresses IP publiques Internet puisque la non existence d'une résolution inverse peut entrainer le refus d'accès à un service. Pour exemple, un serveur de messagerie électronique se présentant en envoi avec une adresse IP n'ayant pas de résolution inverse (PTR) a de grandes chances de se voir refuser, par l'hôte distant, la transmission du courrier (message de refus de type : 'IP lookup failed').

Plusieurs noms de domaine peuvent pointer vers une même adresse IP (via les enregistrements A d'IPv4 ou AAAA d'IPv6).

Réciproquement, une adresse IP peut être résolue en différents noms de domaine via l'enregistrement de plusieurs entrées PTR dans le sous-domaine arpa. dédié à cette adresse (in-addr.arpa. pour IPv4 et in6.arpa. pour IPv6). L'utilisation d'enregistrements PTR multiples pour une même adresse IP est notamment présente dans le cadre de l'hébergement virtuel de multiples domaines web derrière la même adresse IP.

c) Technique du DNS *Round-Robin*

Lorsqu'un service génère un trafic important, celui-ci peut faire appel à la technique du *DNS Round-Robin* (en français tourniquet), qui consiste à associer plusieurs adresses IP à un nom de domaine. Les différentes versions de Wikipedia, comme *fr.wikipedia.org* par exemple, sont associées à plusieurs adresses IP : 207.142.131.235, 207.142.131.236, 207.142.131.245, 207.142.131.246, 207.142.131.247 et 207.142.131.248. Une rotation circulaire entre ces différentes adresses permet ainsi de répartir la charge générée par ce trafic important, entre les différentes machines, ayant ces adresses IP. Il faut cependant nuancer cette répartition car elle n'a lieu qu'à la résolution du nom d'hôte et reste par la suite en cache sur les différents *resolvers* (client DNS).

d) Fully Qualified Domain Name

Les noms d'hôtes sont identifiés de manière unique grâce à leur FQDN (*Fully Qualified Domain Name*, ou Nom de Domaine Pleinement Qualifié). Ils ont le format hôte.domaine.tld. où hôte correspond au nom d'hôte de la machine et domaine.tld. au domaine auquel l'hôte appartient (tld signifie ici *Top Level Domain*, c'est-à-dire l'ensemble des domaines situés directement sous la racine -root .- comme .fr. .com. ou bien .org.). *fr.wikipedia.org.*, par exemple, est composé du domaine générique *org*, du domaine déposé *wikipedia* et du nom d'hôte *fr*.

Le point final, facultatif dans la plupart des commandes, est indispensable en ce qui concerne le DNS. Ainsi, pour *pinguer* une machine ayant pour FQDN machine.domaine.tld., utiliser la commande « ping machine.domaine.tld » ne pose pas de problème, même si le FQDN est incomplet ; toutefois utiliser l'adresse avec le point final « ping machine.domaine.tld. » est plus juste, mais produit un résultat identique. Ainsi, taper http://fr.wikipedia.org. à la place du plus classique http://fr.wikipedia.org dans la barre d'adresse des navigateurs web ne fait aucune différence, car avant d'effectuer la requête DNS, l'implémentation de la pile TCP/IP sous-jacente se charge de rajouter le point final nécessaire à la résolution de nom. À l'inverse, omettre le point final peut avoir des conséquences importantes avec

certaines versions de BIND : spécifier dans le fichier de la zone domaine.tld. que l'hôte machine.domaine.tld a pour adresse IP 1.2.3.4 (*à l'aide d'un enregistrement A, voir ci-dessous*) revient en fait à spécifier que la machine a pour FQDN machine.domaine.tld.domaine.tld.

3) Un système distribué :

Il existe des centaines de milliers de serveurs DNS dans le monde entier. Chacun n'a en réalité à sa disposition qu'un ensemble d'informations restreint.

Quand un hôte a besoin de résoudre un nom de domaine, il doit connaître l'adresse IP d'un ou plusieurs serveurs de noms *récursifs*, c'est-à-dire qui vont éventuellement faire suivre la requête à un ou plusieurs autres serveurs de noms pour fournir une réponse. Les adresses IP de ces serveurs récursifs sont souvent obtenues via DHCP ou encore configurés *en dur* sur la machine hôte. Les fournisseurs d'accès à Internet mettent normalement à disposition de leurs clients ces serveurs récursifs.

Quand un serveur DNS (par exemple, celui d'un fournisseur d'accès à Internet) doit trouver l'adresse IP de *fr.wikipedia.org*, une certaine communication s'instaure alors avec d'autres serveurs DNS. Tout d'abord, notre serveur demande à des serveurs DNS peu nombreux appelés *serveurs racine* quels serveurs peuvent lui répondre pour la zone *org*. Parmi ceux-ci, notre serveur va en choisir un pour savoir quel serveur est capable de lui répondre pour la zone *wikipedia.org*. C'est ce dernier qui pourra lui donner l'adresse IP de *fr.wikipedia.org*.

Pour optimiser les requêtes ultérieures, la plupart des serveurs DNS (et notamment ceux des fournisseurs d'accès à Internet) font aussi office de *DNS cache* : ils gardent en mémoire la réponse d'une résolution de nom afin de ne pas effectuer ce processus à nouveau ultérieurement.

Un nom de domaine peut utiliser plusieurs serveurs DNS. Généralement, les noms de domaines en utilisent au moins deux : un primaire et au moins un secondaire. L'ensemble des serveurs primaires et secondaires font autorité pour un domaine, c'est-à-dire que la réponse ne fait pas appel à un autre serveur ou à un cache. Les serveurs des fournisseurs d'accès à Internet fournissent des réponses qui ne sont pas nécessairement à jour, à cause du cache mis en place. On parle alors de réponse ne faisant pas autorité (**(en)** *non-authoritative answer*).

Pour trouver le nom de domaine d'une IP, on utilise le même principe. Dans un nom de domaine, la partie la plus générale est à droite : org dans fr.wikipedia.org. Dans une adresse IP, c'est le contraire : 213 est la partie la plus générale de 213.228.0.42. Pour conserver une logique cohérente, on inverse l'ordre des quatre termes de l'adresse et on la concatène au pseudo domaine *in-addr.arpa*. Ainsi, par exemple, pour trouver le nom de domaine de l'adresse IP 91.198.174.2, on résout 2.174.198.91.in-addr.arpa, qui est un pointeur vers rr.knams.wikimedia.org.

Cette architecture garantit au réseau Internet une certaine continuité dans la résolution des noms. Quand un serveur DNS tombe en panne, le bon fonctionnement de la résolution de nom n'est pas remis en cause dans la mesure où des serveurs secondaires

sont disponibles. De plus, le DNS permet de mettre à jour l'adresse IP associée à un nom de domaine dans le monde entier facilement et assez rapidement (un délai de 48 heures est généralement suffisant, en fonction de la configuration du nom de domaine).

4) Principaux enregistrements DNS :

Les principaux enregistrements définis par un DNS sont :

- **A record** ou **address record** qui fait correspondre un nom d'hôte à une adresse IPv4 de 32 bits distribués sur quatre octets ex: 123.234.1.2 ;
- **AAAA record** ou **IPv6 address record** qui fait correspondre un nom d'hôte à une adresse IPv6 de 128 bits distribués sur seize octets ;
- **CNAME record** ou **canonical name record** qui permet de faire d'un domaine un alias vers un autre. Cet alias hérite de tous les sous-domaines de l'original ;
- **MX record** ou **mail exchange record** qui définit les serveurs de courriel pour ce domaine ;
- **PTR record** ou **pointer record** qui associe une adresse IP à un enregistrement de nom de domaine, aussi dit « *reverse* » puisque il fait exactement le contraire du A record ;
- **NS record** ou **name server record** qui définit les serveurs DNS de ce domaine ;
- **SOA record** ou **Start Of Authority record** qui donne les informations générales de la zone : serveur principal, courriel de contact, différentes durées dont celle d'expiration, numéro de série de la zone ;
- **SRV record** qui généralise la notion de **MX record**, standardisé dans la RFC 2782 ;
- **NAPTR record** ou **Name Authority Pointer record** qui donne accès à des règles de réécriture de l'information, permettant des correspondances assez lâches entre un nom de domaine et une ressource. Il est spécifié dans la RFC 3403 ;
- **TXT record** permet à un administrateur d'insérer un texte quelconque dans un enregistrement DNS (par exemple, cet enregistrement était utilisé pour implémenter la spécification *Sender Policy Framework*) ;
- d'autres types d'enregistrements sont utilisés occasionnellement, ils servent simplement à donner des informations (par exemple, un enregistrement de type **LOC** indique l'emplacement physique d'un hôte, c'est-à-dire sa latitude et sa longitude).

a) PTR record

À l'inverse d'une entrée de type A, une entrée PTR indique à quel nom d'hôte correspond une adresse IPv4. Si elle est spécifiée, elle doit contenir l'enregistrement inverse d'une entrée DNS A. Par exemple, cet enregistrement PTR :

51.51.51.62.in-addr.arpa IN PTR 3E333333.dslaccess.aol.com

correspond à cette entrée A :

3E333333.dslaccess.aol.com IN A 62.51.51.51

Les enregistrements PTR sont aussi utilisés pour spécifier le nom d'hôte correspondant à une adresse IPv6. Ces entrées de type PTR sont enregistrées dans la zone ip6.arpa., pendant de la zone in-addr.arpa. des adresses IPv4.

La règle permettant de retrouver l'entrée correspondant à une adresse IPv6 est similaire à celle pour les adresses IPv4 (renversement de l'adresse et recherche dans un sous-domaine dédié de la zone arpa.), mais diffère au niveau du nombre de bits de l'adresse utilisés pour rédiger le nom du domaine où rechercher le champ PTR : là où pour IPv4 le découpage de l'adresse se fait par octet, pour IPv6 c'est un découpage par quartet qui est utilisé.

Par exemple, à l'adresse :

4321:0:1:2:3:4:567:89ab

correspond le nom de domaine :

b.a.9.8.7.6.5.0.4.0.0.0.3.0.0.0.2.0.0.0.1.0.0.0.0.0.0.0.1.2.3.4.ip6.arpa.

b) MX record

Une entrée DNS MX indique les serveurs SMTP à contacter pour envoyer un courriel à un utilisateur d'un domaine donné. Sous Unix on peut récupérer les entrées MX correspondant à un domaine à l'aide du programme host(1) (entre autres). Par exemple :

```
$ host -v -t MX wikimedia.org
[...]
;; QUESTION SECTION:
;wikimedia.org.                 IN      MX

;; ANSWER SECTION:
wikimedia.org.          3600    IN      MX      10
mchenry.wikimedia.org.
wikimedia.org.          3600    IN      MX      50
lists.wikimedia.org.
```

On voit que les courriels envoyé à une adresse en @wikimedia.org sont en fait envoyés au serveur mchenry.wikimedia.org. ou lists.wikimedia.org.. Le nombre précédant le serveur représente la priorité. Normalement on est censé utiliser le serveur avec la priorité numérique la plus petite. Ici, c'est donc mchenry.wikimedia.org. qui doit être utilisé en priorité avec une valeur de 10.

Les entrées MX sont rendues obsolètes par les entrées SRV qui permettent de faire la même chose mais pour tous les services, pas juste SMTP (le courriel). L'avantage des entrées SRV par rapport aux entrées MX est aussi qu'elles permettent de choisir un port arbitraire pour chaque service ainsi que de faire de la répartition de charge plus efficacement. L'inconvénient c'est qu'il existe encore peu de programmes clients qui gèrent les entrées SRV.

c) NAPTR record

Peu répandus à l'heure actuelle (ils sont surtout utilisés par ENUM). Ils décrivent une réécriture d'une **clé** (un nom de domaine) en URI. Par exemple, dans ENUM, des enregistrements NAPTR peuvent être utilisés pour trouver l'adresse de courrier électronique d'une personne, connaissant son numéro de téléphone (qui sert de clé à ENUM).

d) SOA record

Cet enregistrement permet d'indiquer le serveur de nom faisant autorité, un contact technique et des paramètres d'expiration. Ces paramètres sont dans l'ordre :

1. **Serial** : indique un numéro de version pour la zone ; ce nombre doit être incrémenté à chaque modification du fichier zone ; on utilise par convention une date au format yyyymmddhhmm ;
2. **Refresh** : le nombre de secondes entre les demandes de mise à jour réalisées depuis le serveur secondaire ou les serveurs esclaves ;
3. **Retry** : le nombre de secondes que doivent attendre le serveur secondaire ou les serveurs esclaves lorsque leur précédente requête a échouée ;
4. **Expire** : le nombre de secondes après laquelle la zone est considérée comme gelée si le secondaire ou les esclaves ne peuvent joindre le serveur primaire ;
5. **Minimum** : utilisé pour déterminer la durée de vie minimum du fichier de zone.

Exemple d'une entrée SOA

```
maboite.com.    IN SOA  serveur.example.com contact.example.com (
                        200612301905  ;serial (version)
                        3600          ;refresh period
                        900           ;retry refresh this often
                        604800        ;expiration period
                        3600          ;minimum TTL
                        )
```

Les versions récentes de BIND (*named*) acceptent les suffixes M, H, D ou W pour indiquer un intervalle de temps en minutes, heures, jours ou semaines respectivement.

II- Exemple de Serveur DNS; BIND :

Cet article a pour but de vous présenter comment installer et configurer un serveur DNS en utilisant l'application **bind9**. Je supposerai que vous disposez d'un réseau local en état de marche et que vous connaissez les bases de TCP/IP (adressage, sous-réseaux,…).

Dans cet article, je vais vous présenter un cas concret de configuration d'un DNS, à vous de l'adapter à vos besoins.1fsBas du formulaire

1) Introduction :

Sur Internet, toutes les machines sont identifiées (et identifiables) par une adresse IP. Cependant, il n'est pas évident de demander à tout-un-chacun de retenir l'adresse IP du serveur web de Google ou plus important encore, du serveur wiki de Ubuntu-fr.

C'est pour cela que l'on a créé les noms de domaines. Les noms de domaine permettent d'identifier un réseau. En ajoutant le nom de la machine, on obtient le nom de l'hôte (hôte se trouvant dans un domaine).

Par exemple, cette page se trouve sur une machine qui est elle-même dans un domaine. Si vous examinez la barre URL de votre navigateur, vous verrez une adresse de ce type :

```
http://doc.ubuntu-fr.org/bind9
```

La partie qui nous intéresse est **doc.ubuntu-fr.org**. Cette chaîne de caractère signifie que vous vous adressez à la machine **doc** qui se trouve sur le domaine **ubuntu-fr.org**.

Lorsque vous introduisez ce nom d'hôte, il est converti en adresse IP afin de pouvoir demander la page `serveur/bind9` au travers du protocole `http`.

L'acronyme DNS signifie *Domain Name System*; en français, *système de nom de domaine*.

Donc, quand votre machine (à la maison ou au bureau) demande le serveur `doc.ubuntu-fr.org`, il s'adresse tout d'abord aux DNS mondiaux pour savoir quelle est la machine qui gère les noms sur le domaine `ubuntu-fr.org`. Imaginons que cette machine se nomme `ns.ubuntu-fr.org`.

Lorsque votre machine sait que `ns.ubuntu-fr.org` gère le nom de domaine `ubuntu-fr.org`, elle interroge le serveur de nom `ns` pour obtenir l'IP de la machine `doc` qui se trouve sur son domaine. A ce moment-là, `ns.ubuntu-fr.org` répond que la machine `doc.ubuntu-fr.org` porte l'adresse IP `212.27.33.233`.

Voilà, comment fonctionne un DNS sans entrer dans les détails.

pour installer un serveur DNS, nous allons utiliser une application bien connue des administrateurs réseaux : **BIND**.

---

2) Installation de BIND :

Pour installer *BIND* sur Ubuntu, installez les paquets **apt://bind9 apt://bind9-doc**

3) Configuration de BIND :

Considérons les aspects suivants :

- `Le réseau local est 192.168.251.* et se nomme bureau.lan.`

- La machine serveur DNS est aussi le serveur de mail et porte l'IP 192.168.251.202; elle se nomme mail2.
- Il y a 3 autres machines sur le réseau : 192.168.251.200 (nommée twin1), 192.168.251.201 (nommée twin2) et 192.168.251.205 (nommée portable).

Remarque : L'utilisation du TLD (*Top Level Domain*) fictif `.lan` est voulue. En effet, n'utilisez pas un TLD existant comme `.com` ou `.ca` sans en être le propriétaire.

Voyons comment configurer le serveur BIND avec ce petit réseau.

Configuration de base du serveur:

Le fichier de configuration générale

La configuration principale de BIND se fait dans le fichier `/etc/bind/named.conf`.

Si vous voulez ajouter vos propres zones, il faut le faire dans le fichier `/etc/bind/named.conf.local`

Dans ce fichier, on définit un certain nombre de *zones*. Une *zone* correspond soit à une plage IP d'un réseau ou à un nom de domaine. Les deux zones qui nous intéressent ici sont `192.168.251.*` et `bureau.lan`.

On définit deux zones pour avoir la résolution de nom dans les deux sens. C'est-à-dire que l'on peut obtenir une adresse IP à partir d'un nom d'hôte mais également, que l'on peut obtenir un nom d'hôte à partir d'une adresse IP.

Une zone avec un nom de domaine se définit comme ceci :

```
zone "bureau.lan" {
        type master;
        file "/etc/bind/db.bureau.lan";
};
```

On indique tout d'abord le nom de la zone que l'on connaît avec le mot clé `zone` suivi du nom de domaine (dans notre cas, `"bureau.lan"`. On indique que c'est le DNS maître (en effet, on peut avoir un ou des DNS de backups qui sont aussi appelés des DNS secondaires) en indiquant `type master`. Et enfin, on indique dans quel fichier se trouve les informations de résolution concernant cette zone. En général, on place ces fichiers dans `/etc/bind/` et on préfixe le nom de la zone par `db.`.

Nous définissons également la zone de plage IP pour la résolution inverse. Pour se faire, nous utilisons les mêmes paramètres. Cependant, le nom de la zone s'écrit avec la plage réseau **inversée** suivi de `.in-addr.arpa`. L'entrée de zone pour notre réseau `192.168.251.*` s'écrit comme ceci :

```
zone "251.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.192.168.251";
};
```

Nous en avons fini avec le fichier de configuration générale. Voyons maintenant comment définir les noms des machines présentes dans une zone.

Les fichiers zones

Comme vous vous en doutez, nous avons un fichier par zone. Les fichiers zones contiennent toutes les entrées comme une table de traduction pour les noms des machines d'une même zone.

Un fichier zone commence toujours par un champ `SOA`, ce champ SOA se compose comme suit :

```
$TTL 3h
@       IN      SOA     ns.bureau.lan. hostmaster.bureau.lan. (
                                2005090201
                                8H
                                2H
                                1W
                                1D )
```

Le symbole `@` désigne la zone décrite par le fichier de configuration (ici, `bureau.lan`). A la place de `@`, vous auriez très bien pu indiquez `bureau.lan.` (*n'oubliez pas le "." à la fin !!!*).

Ensuite, on indique `IN` qui signifie que l'on a affaire à une zone Internet; c'est pour ainsi dire toujours le cas (sauf quelques très rares exceptions). Enfin, toujours sur la première ligne, on indique le serveur DNS qui dispose du fichier zone de référence (important lorsque que l'on a des DNS secondaires) et l'adresse email de la personne responsable de la zone (le premier "." dans le champ d'email est considéré comme un "@").

Dans notre cas, le serveur DNS primaire de la zone est `ns.bureau.lan` et l'adresse email de la personne responsable est `hostmaster@bureau.lan`.

Remarque : Vous avez sans doute noté que le serveur DNS et l'adresse email sont ponctuées par un point ("."). Ce point est **indispensable**. Si vous l'omettez, par défaut, BIND rajoute le nom de la zone et dès lors `ns.bureau.lan.` renvoie `ns.bureau.lan` alors que `ns.bureau.lan` (sans point) renvoie `ns.bureau.lan.bureau.lan`. Il s'agit d'une erreur très fréquente.

Les valeurs qui suivent sont respectivement :

- `le numéro de série (souvent on met la date courante suivie d'un numéro d'ordre); AAAAMMJJxx.`
- `le temps de rafraichissement (refresh; ici, 8 heures); la valeur recommandée est de 24 heures.`
- `le temps entre deux essais (retry; ici, 2 heures); la valeur recommandée est de 2 heures.`
- `le temps d'expiration (expire; ici, 1 semaine); la valeur recommandée est de 1000 heures.`
- `la valeur TTL minimum (minimum; ici, 1 jour); la valeur recommandée est de 2 jours.`

En utilisant les valeurs que j'ai stipulées ci-dessus, tout devrait fonctionner.

Après le champ SOA, on indique le serveur de nom à consulter pour résoudre un nom d'hôte dans le domaine `bureau.lan`. Nous faisons ça avec un champ `NS` de la manière suivante :

```
@       IN      NS      ns.bureau.lan.
```

Ensuite (ceci est facultatif), si vous avez un serveur de mail, vous pouvez indiquer au serveur DNS que les adresses de la forme `*@bureau.lan` sont gérées par un serveur de mail prédéfini; nous le faisons comme ceci :

```
@       IN      MX      10      mail2.bureau.lan.
```

Remarque : La valeur `10` indique la priorité du serveur concerné. En indiquant plusieurs champs `MX` avec des valeurs différentes, vous pouvez indiquer des serveurs de mail secondaires.

Enfin, nous terminons ce fichier zone avec *la table de traduction* des hôtes en adresse IP :

```
ns              A       192.168.251.202
mail2           A       192.168.251.202
twin1           A       192.168.251.200
twin2           A       192.168.251.201
portable        A       192.168.251.205
```

Le fichier zone complet pour `bureau.lan` ressemble à ceci :

```
$TTL 3h
@       IN      SOA     ns.bureau.lan. hostmaster.bureau.lan. (
                                2005090201
                                8H
                                2H
                                1W
                                1D )

@       IN      NS      ns.bureau.lan.

@       IN      MX      10      mail2.bureau.lan.


ns              A       192.168.251.202
mail2           A       192.168.251.202
twin1           A       192.168.251.200
twin2           A       192.168.251.201
portable        A       192.168.251.205
```

Avant de pouvoir utiliser notre serveur DNS, nous allons renseigner la zone pour la plage IP de notre réseau. La zone se décrit vaguement comme la précédente, à la différence près que l'on utilise le mot clé `PTR` au lieu de `A` dans la table de traduction.

Voici le fichier zone pour notre réseau 192.168.251.* d'exemple :

```
$TTL 3h
@       IN      SOA     ns.bureau.lan. hostmaster.bureau.lan. (
                                2005090201
                                8H
                                2H
                                1W
                                1D )

@       IN      NS      ns.bureau.lan.

@       IN      MX      10      mail2.bureau.lan.

$ORIGIN 251.168.192.in-addr.arpa.
202     IN      PTR     ns.bureau.lan.
202     IN      PTR     mail2.bureau.lan.
200     IN      PTR     twin1.bureau.lan.
201     IN      PTR     twin2.bureau.lan.
205     IN      PTR     portable.bureau.lan.
```

Si la ligne $ORIGIN 251.168.192.in-addr.arpa. vous crée une erreur (voir le fichier log /var/log/daemon.log apres avoir redemarré bind9) vous pouvez la supprimer, cela fonctionne

Vérification de la configuration :

Pour vérifier le fonctionnement de notre serveur DNS, nous allons utiliser les utilitaires fournit avec BIND9 **named-checkconf named-checkzone**.

Vérification de la configuration global named-checkconf

Le programme named-checkconf sert à vérifier la syntaxe du fichier /etc/bind/named.conf.

Version de named-checkconf :

```
named-checkconf -v
```

Contrôle du fichier :

```
named-checkconf /etc/bind/named.conf
```

Vérification de la configuration des zones named-checkzone

Le programme named-checkzone vérifie la syntaxe et la cohérence d'un fichier de zone maître.

```
named-checkzone nom_zone fichier_de_zone
```

Nous pouvons maintenant demander à notre serveur de prendre en compte nos modifications en rechargeant la configuration de bind :

```
sudo /etc/init.d/bind9 reload
```

Nous pouvons maintenant passer à la phase de vérification.

Pour vérifier le fonctionnement de notre serveur DNS, nous allons lui adresser des requêtes directement via l'utilitaire `nslookup`, pour l'utiliser, il suffit de taper `nslookup` dans un terminal.

On doit lui indiquer le serveur DNS à vérifier via le mot clé `server 127.0.0.1` et ensuite, on lui donne un nom d'hôte et il doit nous répondre l'adresse IP.

Voici la petite session de test que j'ai fait chez moi :

```
nslookup
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
> mail2.bureau.lan
Server:         127.0.0.1
Address:        127.0.0.1#53

Name:   mail2.bureau.lan
Address: 192.168.251.202
> 192.168.251.201
Server:         127.0.0.1
Address:        127.0.0.1#53

201.251.168.192.in-addr.arpa    name = twin2.bureau.lan.
> set q=mx
> bureau.lan
Server:         127.0.0.1
Address:        127.0.0.1#53
bureau.lan    mail exchanger = 10 mail2.bureau.lan.
> exit
```

Si tout se déroule normalement, vous pouvez configurer vos clients et utiliser votre serveur DNS.

Configuration avancée :

Configuration d'un serveur DNS secondaire

Dans cette section, nous allons envisager la configuration d'un serveur DNS secondaire qui se synchronise avec le serveur DNS principal que nous avons défini ci-dessus.

Le serveur DNS secondaire proprement dit

Pour configurer le serveur secondaire, nous devons simplement indiquer à BIND les zones qu'il doit traiter en mode esclave. Sur base de la configuration ci-dessus, nous configurons le fichier `/etc/bind/named.conf` de serveur DNS secondaire de la manière suivante :

```
zone "bureau.lan" {
        type slave;
        masters {192.168.251.202;} ;
        file "/etc/bind/db.bureau.lan";
};
```

```
zone "251.168.192.in-addr.arpa" {
        type slave;
        masters {192.168.251.202;} ;
        file "/etc/bind/db.192.168.251";
};
```

En faisant cela, il est inutile d'indiquer les fichiers zones (fichier `db.`) sur le serveur DNS secondaire. Les fichiers proviendront d'une synchronisation avec le DNS primaire.

//Remarque :// L'utilisateur faisant fonctionner le serveur DNS doit avoir les droits d'écriture sur les fichiers zones renseignés dans la configuration ci-dessus.

Modification de la configuration du serveur DNS primaire

Nous devons renseigner dans les fichiers zones le deuxième serveur DNS, et pour se faire, on ajoute la ligne suivante au fichier `/etc/bind/db.bureau.lan` :

```
@       IN      NS      ns2.bureau.lan.
```

et nous devons également renseigner l'adresse IP de `ns2.bureau.lan` (n'oubliez pas de mettre à jour le fichier de zone pour le sous-réseau IP également avec le mot clé `PTR` !) :

```
ns2             IN      A       192.168.251.250
250             IN      PTR     ns2.bureau.lan.
```

Après avoir modifié les zones et de ce fait, dévoilé `ns2`, nous pouvons maintenant indiquer au serveur DNS maître que le serveur DNS secondaire peut accéder aux données de zones.

Pour ce faire, les informations concernant les zones qui nous intéressent (dans le fichier `/etc/bind/named.conf` du serveur maître) deviennent ceci :

```
zone "bureau.lan" {
        type master;
        notify yes;
        allow-transfer {192.168.251.250;} ;
        file "/etc/bind/db.bureau.lan";
};

zone "251.168.192.in-addr.arpa" {
        type master;
        notify yes;
        allow-transfer {192.168.251.250;} ;
        file "/etc/bind/db.192.168.251";
};
```

Après toutes ces modifications, demandez au service BIND de recharger la configuration :

```
sudo /etc/init.d/bind9 reload
```

Et surtout, n'hésitez pas à re-tester votre configuration (sur les deux serveurs).

4) Configuration des clients :

La configuration de la résolution de nom pour les machines Linux se fait dans le fichier `/etc/resolv.conf`. Dans ce fichier, vous pouvez ajouter le domaine de recherche via la ligne suivante (en premier dans le fichier) :

```
search bureau.lan
```

Et ensuite, les adresses de vos serveurs de noms (primaire interne, autres internes, puis ceux de votre fournisseur d'accès par exemple) de la manière suivante :

```
nameserver 192.168.251.202
```

L'ordre dans lequel vous indiquez les lignes est important. Voici le fichier tel qu'il est chez moi :

```
search bureau.lan
nameserver 192.168.251.202
nameserver 192.168.251.212
nameserver 193.121.171.135
nameserver 193.74.208.65
```

Linux va essayer de résoudre un nom de la manière suivante (si une étape ne fonctionne pas, il essaye la suivante) :

- recherche du serveur de nom de bureau.lan et interrogation de ce serveur.
- interrogation du serveur DNS 192.168.251.202 qui est mon serveur DNS primaire (interne).
- interrogation du serveur DNS 192.168.251.212 qui est mon serveur DNS secondaire (interne).
- interrogation du serveur DNS 193.121.171.135 qui est le serveur DNS primaire de mon provider.
- interrogation du serveur DNS 193.74.208.65 qui est le serveur DNS secondaire de mon provider.

Configuration des clients Windows

Sans entrer dans les détails, il vous suffit d'introduire l'adresse IP de vos serveurs DNS primaire et secondaire dans les propriétés du protocole TCP/IP (accessible dans les connexions réseaux du panneau de configuration). Rajouter le suffixe DNS correspondant à votre domaine dans les propriétés réseaux de votre carte.

III- Sécurité du DNS :

Comme beaucoup de protocoles Internet, le DNS a été conçu sans se préoccuper de la sécurité. Il ne faut donc pas se fier au DNS pour arriver sur le bon serveur et c'est pour cela que des protocoles comme SSH font leur propre vérification (via la cryptographie). Les principales failles du DNS (décrites dans le RFC 3833) sont :

- l'interception du paquet (requête ou réponse) et émission d'un autre paquet à sa place ;
- la fabrication d'une réponse (les serveurs DNS acceptent trop facilement des réponses puisque seul un numéro de requête, très petit, sert d'authentification) ;
- la trahison par un serveur (le secondaire hors-site d'un domaine peut par exemple passer sous le contrôle de personnes malintentionnées) ou corruption de données ;
- l'empoisonnement du cache DNS ;
- le déni de service (saturation du serveur par un grand nombre de requêtes simultanées).

1) DNSSEC :

Pour contrer ces vulnérabilités, le protocole DNSSEC a été développé.

DNSSEC (abréviation de DNS Security Extensions) ajoute à la sécurité du Domain Name System.

DNSSEC a été conçu pour protéger l'Internet à partir de certaines attaques, comme DNS cache poisoning [0]. Il s'agit d'un ensemble d'extensions au DNS, qui prévoient: une authentification) de l'origine des données DNS, b) l'intégrité des données, et c) le refus authentifiée de l'existence.

Ces mécanismes nécessitent des modifications du protocole DNS. DNSSEC ajoute quatre nouveaux types d'enregistrement de ressource: Resource Record Signature (RRSIG), DNS à clé publique (dnskey), délégation Signer (DS), et la prochaine (Secure NSEC). Ces enregistrements de ressources nouvelles sont décrites en détail dans la RFC 4034.

Elle ajoute aussi deux nouveaux indicateurs d'en-tête DNS: Vérifier CD (handicapés) et authentifié de données (AD). Afin de soutenir les tailles de message plus grand DNS qui résulte de l'addition des enregistrements de ressources DNSSEC, DNSSEC exige également EDNS0 soutien (RFC 2671).

Enfin, DNSSEC nécessite le support du DNSSEC OK (DO) EDNS header bit de la tête (RFC 3225) ainsi qu'une garantie-resolver conscients peuvent indiquer, dans ses questions qu'il souhaite recevoir DNSSEC RI dans des messages de réponse. En cochant la signature, un résolveur DNS est en mesure de vérifier si l'information est identique (correcte et complète) de l'info sur le serveur DNS faisant autorité.

DNSSEC services de protection contre la plupart des menaces pesant sur le Domain Name System. Il existe plusieurs catégories distinctes de menaces pour le Domain Name System, dont la plupart sont liés à DNS cas de problèmes plus généraux, mais dont quelques-unes sont spécifiques aux particularités du protocole DNS.

Notez que DNSSEC ne prévoit pas la confidentialité des données. Par ailleurs, DNSSEC ne protège pas contre les attaques de DDoS.

---

[0] Une analyse complète de la menace du Domain Name System peut être trouvé dans la RFC 3833. Cette RFC tente de décrire certaines des menaces connues du DNS, et - ce faisant - vise à mesurer dans quelle mesure DNSSEC est un outil utile à la défense contre ces menaces.

Le noyau de la spécification DNSSEC est décrit dans les 3 suivants RFC, publiés Mars 2005:

RFC 4033 - DNS Security Introduction and Requirements

RFC 4034 - Resource Records for the DNS Security Extensions

RFC 4035 - Protocol Modifications for the DNS Security Extensions

RFC 5155 (Mars 2008) introduit un enregistrement de ressources de substitution, NSEC3, qui prévoit des mesures supplémentaires contre la zone de recensement et les permis de l'expansion progressive de la délégation de zones centrée.

RFC 5155 - DNS Security (DNSSEC) Hashed Authenticated Denial of Existence

RFC connexes, tels que la RFC 4310, décrivent comment la carte DNSSEC pour la Extensible Provisioning Protocol (EPP). RFC 4641 décrit DNSSEC pratiques opérationnelles.

RFC 4310 - Domain Name System (DNS) Security Extensions Mapping for the Extensible Provisioning Protocol (EPP)

RFC 4641 - DNSSEC Operational Practices

DNSSEC de gestion de clés, y compris la pièce de roulement, se fait en utilisant un logiciel spécialisé DNSSEC, qui peuvent constituer des outils autonomes ou add-ons à votre logiciel de DNS existante. Tous les principaux logiciels DNS aura plein ou partiel DNSSEC fonctionnalité intégrée dans les prochaines années.

Pour rendre plus facile le déploiement de DNSSEC, on peut aussi acheter un dédié "DNSSEC Appliance», qui agit en tant que signataire DNS automatique pour les zones DNS. Plusieurs fournisseurs offrent déjà des commerciaux et non des solutions commerciales pour la signature de DNS en temps réel, certains d'entre eux en utilisant le matériel de chiffrement externes tels que HSM (Hardware Security Modules), y compris les jetons USB et les cartes à puce.

2) DNSSEC Notes :

Le présent document résume l'état de la mise en œuvre dans DNSSEC
cette version de Bind9.

OpenSSL librairie requis

À l'appui de DNSSEC, BIND 9 doit être liée avec la version 0.9.6e ou plus récente de la bibliothèque OpenSSL. Comme de BIND 9.2, la bibliothèque n'est plus inclus dans la distribution - il doit être fourni par l'exploitation système ou installé séparément.

Pour construire BIND 9 avec OpenSSL, utilisez "configure - with-openssl». Si la bibliothèque OpenSSL est installé dans un emplacement non standard, vous pouvez spécifier un chemin comme dans "configure - with-openssl = / var".

Génération de la clé et de signature

Les outils de génération des clés DNSSEC et les signatures sont maintenant dans la / bin dnssec répertoire. Documentation de ces programmes peuvent être consultés doc/arm/Bv9ARM.4.html et dans les pages man.

Les données aléatoires utilisés pour générer les clés et les signatures vient DNSSEC soit à partir de / dev / random (si l'OS le supporte) ou le clavier.
Alternativement, un dispositif ou un fichier contenant entropie / données aléatoires peuvent être
spécifiée.

Desservant les zones sécurisées

Lorsqu'il agit comme un serveur de noms faisant autorité, BIND9 comprend KEY, SIG
et NXT dossiers dans les réponses comme spécifié dans RFC2535 lorsque la demande a le drapeau Ne définissez dans la requête.

Secure Résolution

Support de base pour la validation des signatures dans les réponses a DNSSEC été mises en œuvre, mais devrait être encore considéré comme expérimental.

Lorsqu'il agit comme un serveur de noms cache, BIND9 est capable de réaliser base de validation DNSSEC des résultats positifs ainsi que les réponses non-existence.
Cette fonctionnalité est activée en incluant un «trusted-keys" clause dans le fichier de configuration, contenant la clé de haut niveau de la zone l'arbre DNSSEC.

Validation des réponses génériques n'est pas actuellement prise en charge. Dans particulier, un nom "n'existe pas" réponse validera
succès, même si elle ne contient pas les enregistrements NXT pour prouver l'inexistence d'un caractère générique correspondant.

Preuve du statut de précarité pour les zones d'insécurité délégués d'être sécurisé

zones fonctionne lorsque les zones sont complètement non sécurisé. Privée
Zones sécurisées délégués par zones sécurisées ne fonctionnent pas dans tous les cas, par exemple lorsque la zone privée garanti est desservi par le même serveur
comme un ancêtre (mais pas de mère) de zone.

La manipulation du bit de CD dans les requêtes est maintenant pleinement mis en œuvre. Validation
On ne tentera pas de requêtes récursives si le CD est défini.

Secure Dynamic Update

Mise à jour dynamique de la zone sécuritaire a été mis en œuvre, mais mai ne pas être complète. Visé NXT et enregistrements SIG sont mises à jour par le serveur lorsque une mise à jour se produit. Avancées de contrôle d'accès est possible en utilisant le "actualiser la politique" présenté dans la définition de zone.

Secure Zone Transferts

BIND 9 ne met pas en œuvre les mécanismes de transfert de zone de sécurité RFC2535 section 5.6, et nous n'avons pas prévu de les mettre en œuvre dans le avenir comme nous les considérons comme inférieurs à l'utilisation des TSIG ou SIG (0) à assurer l'intégrité des transferts de zone.

3) Autre méthodes de Sécurit :

* Listes de contrôle d'accès
Access Control Lists (ACL), sont des listes qui correspondent à l'adresse que vous pouvez mettre en place et un surnom pour une utilisation future
allow-notify, allow-query, allow-query-on, allow-recursion, allow-recursion-on, blackhole, allowtransfer,
etc

* allow-query

```
  allow-query { address_match_list };
```

allow-query définit une liste de correspondance par exemple, IP address (es) qui sont autorisées à émettre des requêtes vers le serveur. Si non spécifié, tous les hôtes sont autorisés à faire des requêtes. Cette déclaration mai être utilisé dans une vue ou une clause d'options globales.

* allow-recursion

```
allow-recursion { address_match_list };
```

allow-recursion définit une liste de correspondance par exemple, IP address (es) qui sont autorisées à émettre des requêtes récursives sur le serveur. Si la réponse à la requête existe déjà dans le cache, il sera retourné indépendamment de la présente

déclaration. Si non spécifié, tous les hôtes sont autorisés à effectuer des requêtes récursives. Cette déclaration peut être utilisé dans une vue ou une clause d'options globales.

*blackhole

```
blackhole { address_match_list };
```

Blackhole définit un address_match_list des armées que le serveur ne sera pas répondu, ou des requêtes en répondre. Le défaut est 'none' (tous les hôtes reçoivent une réponse). Cette déclaration mai seulement être utilisée dans une clause des options globales.

Utilisation de listes de contrôle d'accès vous permet d'avoir un contrôle sur qui peut accéder à votre serveur de nom, sans l'encombrer
vos fichiers de configuration avec des listes d'adresses IP.
C'est une bonne idée d'utiliser les ACL, et de contrôler l'accès à votre serveur. Limiter l'accès à votre serveur des parties extérieures peut aider à prévenir l'usurpation d'identité"spoofing" et le déni de service (DoS) attaque contre votre serveur.
Voici un exemple de la façon d'appliquer les ACL:
/ / Mettre en place une ACL nommé "bogusnets" qui permet de bloquer l'espace RFC1918
/ / Et un espace réservé, qui est couramment utilisé dans les attaques de "spoofing".
acl {bogusnets
0.0.0.0 / 8; 1.0.0.0 / 8; 2.0.0.0 / 8; 192.0.2.0/24; 224.0.0.0 / 3;
10.0.0.0 / 8, 172.16.0.0/12, 192.168.0.0/16;
};
/ / Mettre en place une ACL appelé nos filets. Remplacer avec les vrais numéros IP.
acl nos filets {x.x.x.x/24; x.x.x.x/21;};
options {
...
...
allow-query {nos filets;};
allow-recursion {nos filets;};
...
blackhole {bogusnets;};
...
};
zone "example.com" {
type master;
file "m / example.com";
allow-query {any;};
};
Cela permet de requêtes récursives du serveur de l'extérieur, sauf si recursion a été précédemment désactivé.

* Chroot et Setuid
Sur les serveurs UNIX, il est possible de faire fonctionner BIND dans un environnement chrooté (en utilisant le chroot () fonction) par

précisant les option "-t". Cela peut aider à améliorer la sécurité du système par la mise BIND dans un "sandbox»,
qui limite les dégâts, si un serveur est compromise.
Une autre fonction utile dans la version UNIX de BIND est la capacité à exécuter le démon en tant que démunis
utilisateur (-u user). Nous suggérons de fonctionner comme un utilisateur sans privilèges lors de l'utilisation de la fonction chroot.
Voici un exemple de ligne de commande pour charger BIND dans un bac à sable chroot, / var / named, et nommé à courir
setuid 202 à l'utilisateur:
/ usr / local / bin / named-u 202-t / var / named
-De l'environnement chroot
Pour un environnement chroot pour fonctionner correctement dans un répertoire particulier (par exemple, / var /
nom), vous aurez besoin de mettre en place un environnement qui comprend tout ce dont a besoin pour exécuter BIND. À partir de
BIND, du point de vue, / var / named est la racine du système de fichiers. Vous aurez besoin d'ajuster les valeurs de
comme des options comme le répertoire et pid-file pour tenir compte de cela.
Contrairement aux précédentes versions de BIND, en général vous n'aurez pas besoin de compiler statiquement nom ni installer
bibliothèques partagées dans le cadre de la nouvelle racine. Toutefois, en fonction de votre système d'exploitation, vous avez besoin de mettre en place des choses comme / dev / zero, / dev / random, / dev / log et / etc / localtime.
- Utilisation de la fonction setuid
Avant de lancer le démon du nom, utilisez la touche d'utilité (pour changer des fichiers d'accès et de modification de fois)
chown ou l'utilité (pour définir le nom d'utilisateur et / ou identifiant de groupe) sur les fichiers auxquels vous voulez écrire à BIND.
NOTE
Notez que si le démon du nom est en cours d'exécution comme un utilisateur sans privilèges, il ne sera pas
capable de se lier à de nouveaux ports limité si le serveur est rechargé.
* Mise à jour dynamique de sécurité
L'accès à la facilité de mise à jour dynamique doit être strictement limitée. Dans les anciennes versions de BIND, la seule
façon de faire est basée sur l'adresse IP de l'hôte demandant la mise à jour, par l'inscription d'une adresse IP
préfixe réseau ou en permettre la mise à jour de la zone option. Cette méthode est précaire depuis l'adresse source de
la mise à jour de paquets UDP est facilement falsifié. Notez également que si les adresses IP autorisées par le "allow-update" option comprennent l'adresse d'un serveur esclave, qui effectue la transmission des mises à jour dynamiques, le capitaine peuvent être trivialement attaqué par l'envoi de la mise à jour de l'esclave, qui le transmet à la maîtrise de ses propre adresse IP source, l'origine du maître de l'approuver sans question.
Pour ces raisons, nous recommandons vivement que les mises à jour cryptographiques soit authentifié par les moyens les signatures de transaction (TSIG). C'est permettre à l'option de mise à jour devrait afficher seulement TSIG clé, non pas les adresses IP ou de réseau préfixes. Sinon, la nouvelle mise à jour en matière de politique option peut

être utilisée.
Certains sites choisissent de conserver toutes les mises à jour DNS dynamique des données dans un sous-domaine et de ce sous-domaine délégué à une zone distincte. De cette façon, le haut niveau de la zone contenant des données critiques telles que les adresses IP des Web public et les serveurs de messagerie ne doivent pas permettre la mise à jour dynamique du tout.