*Travaux Pratiques n°4*

# Configuration d'un serveur DNS

But du TP

Ce TP veut aborder la manipulation des notions de *Domain Name Server* en tant que client puis comme serveur dans un contexte simple : pas de sécurité ( DNSSEC ), pas de DNS dynamique.

## Mise en place

Cette manipulation peut être effectuée sur chacun des postes des salles 410, 405 voire 102 sans notion de poste client et poste serveur. Tous les PC peuvent être à la fois clients et serveurs. Comme vous devez être administrateur sur le système, vous devrez travailler sur une machine virtuelle. Ce TP suppose l'utilisation du disque virtuel `Debian1102-20220323-pm...`, Il faut donc créer une machine virtuelle comme indiqué en adaptant le nom de votre vm.

```
createvm
la commande sans paramètre vous donne la syntaxe à utiliser.
createvm Debian1102-20220323-pm...
```

***Ne pas changer l'adresse MAC***, cela empêcherait la VM d'obtenir une adresse IP prévisible. On suppose aussi qu'un câble droit relie directement l'interface `en1` de votre PC au switch de la salle (SW102, SW410 ou SW405).

Faites démarrer votre VM et logez-vous en root ou en rt.

Dans tout le TP il va être question d'adresses IP et de noms de machines. Les exemples utilisent soit la salle 405, soit la 410, soit la 102 : il convient de tout ramener à la salle dans laquelle vous effectuez la manipulation.

# I. Fonctionnement comme client

### *1. La commande de base*

La commande la plus simple pour utiliser un serveur DNS est `host` :

```
host kheops.unice.fr
```

On obtient une réponse simple qui contient l'adresse IPv4.

```
kheops.unice.fr has address 134.59.136.6
```

On peut jouer dans l'autre sens pour obtenir le nom à partir de l'adresse.

```
host 134.59.136.6
```

Qu'obtient-on ?

```
6.136.59.134.in-addr.arpa domain name pointer kheops.gtr.tp.
```

Ça c'est plus compliqué et mérite des explications.

Tout d'abord la fin : « kheops.gtr.tp. » le nom est complet c'est-à-dire avec la partie domaine (fqdn). On remarque aussi un « . » à la fin qui n'est peut-être pas la fin de la phrase.

Le milieu « domain name pointer » indique simplement une traduction, rien de passionnant sauf que nous aurons tout-à l'heure un mot clé « PTR ».

Maintenant le début. On dirait que tout est mélangé ! Non mais à l'envers oui. Les adresses du DNS Inverse (@IP → nom) utilisent les valeurs de manière inversée. Ça fait bizarre au début mais à l'usage c'est très pratique. On a un pseudo-domaine « in-addr.arpa » celui des « internet address » et le reste est l'adresse 134.59.136.6 dans l'ordre inverse 6.136.59.134.

## *2. La même en plus bavarde*

### a) Première réponse détaillée

Demandons à host de nous dire ce qu'il fait

```
host -v kheops.unice.fr
```

Quelle est l'adresse IP de la machine qui a répondu ? (ligne « Received nn bytes from… »).

Combien de questions ont-elles été posées ? Il y en a **trois**, elles commencent par « Trying "kheops.unice.fr". Une pour le type A, une pour le type AAAA et la troisième pour le type MX.

Pour n'obtenir que l'adresse IPv4 (type A)utilisez :

```
host -v -t A kheops.unice.fr
```

On peut changer de serveur de noms pour maat.unice.fr :

```
host -v -t A kheops.unice.fr maat.unice.fr
```

On observe que la réponse est la même : en fait c'est la même machine qui génère la réponse : taloa.unice.fr mais qui est inaccessible directement à partir des salles de TP.

taloa.unice.fr est la machine qui gère le domaine unice.fr. On a plus d'informations : dans l'entête (header) on voit qu'il a 1 question, 1 réponse, il a trouvé 3 autorités qui auraient pu répondre (type NS) et enfin 3 informations supplémentaires qui sont nécessaires.

Maintenant avec le serveur de noms de l'AD de l'IUT, on obtient quelque chose de différent :

```
host -v -t A kheops.unice.fr dnsAD.gtr.tp
```

Les lignes qui commencent par « ; » sont des commentaires. Elles nous rappellent la question : on veut l'adresse IPv4 (A) de `kheops.unice.fr`.

La réponse suit une syntaxe que l'on va retrouver partout :

```
"objet recherché" TTL Classe Type "données ressources"
```

Analysons :

- Le premier champ est évident
- le champ TTL (TimeToLive = durée de vie) c'est la durée de validité de la réponse en secondes : 86400 fait 24 heures. (facultatif lors de la configuration). Si vous rappelez la même commande `host` plusieurs fois de suite, vous remarquerez que le champ TTL varie. C'est la durée de validité résiduelle de la réponse DNS dans le cache local.
- La classe est quasiment toujours IN : Internet. Ce champ est tellement fréquent qu'il est très souvent omis dans les fichiers de configuration.
- Le type ne peut pas être omis. On retrouve A pour l'adresse IPv4.
- La donnée recherchée est bien une adresse IPv4.

La réponse obtenue par `centre` ou `maat`, comporte d'autres informations. Nous allons les analyser, car on voit là les principaux éléments des fichiers de « zone » qu'il nous faudra remplir dans la deuxième partie.

Dans la section « autorités » on trouve des lignes dont le type est NS pour « Name Server ». Elles indiquent les machines qui sont les serveurs de noms officiels de la zone « `unice.fr.` ». Quels sont-ils ?

Vous notez que la zone (ailleurs on dit plutôt le domaine) finit par un « . » comme le nom des serveurs.

La section additionnelle permet juste d'avoir l'adresse des serveurs de noms sans avoir besoin d'interroger un serveur de noms. C'est obligatoire, vous en aurez besoin dans la suite.

### b) Le serveur ne sait pas tout

Un peu plus haut, notre première commande comportait 3 réponses, étudions un peu plus les deuxième et troisième.

```
host -v kheops.unice.fr
```

La première est analysée, c'est clair ! La seconde demande le champ AAAA, c'est une adresse IPv6 (4x32 bits = 128 bits) quelle est réponse ?

Ensuite c'est le champ MX : le `mail-exchanger` c'est-à-dire le serveur de mail qui prend en charge le mail de `kheops`. Quelle est la réponse ?

Nous avons ici une section d'autorité dont le type est **SOA** (**S**`tart` **O**`f` **A**`utority`) c'est-à-dire les informations sur le serveur qui est responsable de la zone (ici `unice.fr`.).

Les champs « données ressources » ont comme signification

- le serveur (`taloa.unice.fr`)
- l'adresse du responsable de la zone, mais sans « @ » car en DNS ce caractère a une signification particulière (la zone, le domaine).
- un numéro de série qui doit pouvoir être interprété comme une date précise : elle sert à déterminer s'il est nécessaire de mettre à jour les serveurs secondaires.
- une durée (en secondes) avant laquelle il faudrait rafraichir la base sur un serveur de nom secondaire.
- un temps d'attente avant de réessayer si le rafraichissement a échoué
- la durée maximale après laquelle le serveur secondaire n'est plus considéré comme une autorité
- la durée de vie minimale de tous les champs distribués dans cette zone.

Traduire et donner les valeurs sur `unice.fr` sous une forme interprétable.

À la fin de chaque réponse on voit le serveur qui a répondu et le port qui a servi à l'interroger. (différent suivant la question posée).

`134.59.136.1#53`

## *3. Mais comment fait-il ?*

### a) Comportement étrange

C'est assez simple ! Regardons ce qui se passe si l'on interroge mal le serveur.

```
host kheops
```

Normalement le nom « `kheops` » n'existe pas. C'est le nom complet, avec le domaine donc, qui existe de manière **unique** sur Internet. Est-ce que ce serait magique ? Oui et non. Oui, le système a des informations par ailleurs, non, car nous pouvons maîtriser ces informations.

Tentons d'en savoir plus :

```
host -v -t A kheops
```

On voit que même s'il a trouvé la bonne adresse, il y a quelque chose d'étrange sur la ligne « `Trying` ».

Il nous indique qu'il essaie `kheops.gtr.tp`. Surprenant n'est-ce pas ?

```
## Dynamic resolv.conf(5) file for glibc resolver(3) generated by resolvconf(8)
#     DO NOT EDIT THIS FILE BY HAND -- YOUR CHANGES WILL BE OVERWRITTEN
nameserver 10.4.105.251
search gtr.tp
```

*Listing 1: Contenu du fichier `/etc/resolv.conf`*

### b) Le fichier `resolv.conf`

C'est que lorsque le système cherche à interroger un serveur DNS, il va à chaque fois regarder dans le fichier `/etc/resolv.conf`, pour savoir à qui il doit s'adresser et comment.

On voit que le système va utiliser le serveur (10.4.105.251).

La directive `search` indique que le système va essayer de compléter un nom d'hôte court (ne finissant pas par un point « . »), en lui ajoutant la chaîne de recherche. Donc il est logique que le système cherche `kheops` avec `kheops.gtr.tp`. Et comme l'administrateur l'a défini sur le serveur DNS 10.4.105.251, il le trouve et nous revoit son adresse IP.

Que ce passe-t-il si vous passez la commande

```
host -v kheops.gtr
```

Il peut aussi être pratique de lister la zone entière.

```
host -l gtr.tp.
```

C'est long ! Sauriez-vous dire combien la zone `gtr.tp` comporte de noms de machines ? Tous ces noms ne correspondent pas à de vraies machines physiques !

Analysez ce qui se passe dans la commande suivante

```
host -avl gtr.tp. maat
```

### c) Pas d'erreur d'interprétation ?

Dans les réponses précédentes, nous avons peut-être mal lu, car nous avons suivi la logique de la commande host. Elle nous envoie des réponses à des questions non posées. Par exemple dans les réponses obtenues au paragraphe I. 2. b) page 3, certaines lignes commençaient par `unice.fr` le nom de domaine et non `kheops.unice.fr` comme dans la question posée. En effet cette information complémentaire était utile pour bien interpréter les véritables réponses.

Pour mieux appréhender la notion de domaine ( = de zone), au niveau du service de noms, nous allons directement interroger la base au sujet du domaine.

```
host -v -t ANY  unice.fr
```

Le type demandé est « ANY » c'est-à-dire tous les types définis pour `unice.fr`.

Le type ANY est variable : on retrouve ce qui est dans le cache local. On obtient pour une seule question, 4 réponses : les NS et leurs adresses. Pour en avoir plus il faut le demander : testez avec les types TXT, CAA, SOA, MX… Attention, certains types sont pour des machines d'autres pour des zones, la commande ci-dessus ne demande qu pour la zone unice.fr. .

On retrouve les types NS (3 serveurs de noms), le type SOA.
Quel est le serveur de nom primaire de la zone `unice.fr` ?
Quels sont les serveurs de noms secondaires ?

Les deux serveurs de mails de la zone correspondent au type MX avec une priorité différente (10 est mieux que 20)

Les types TXT peuvent être utilisés sans contrainte de normes

Les types CAA désignent :

- `issue` : l'autorité de certification autorisée sur le domaine
- `iodef` : l'action à faire en cas de violation de la contrainte « `issue` » ; ici il faut envoyer un mail à `abuse@unice.fr`.

### *4. La commande `dig`*

La commande `dig` permet aussi d'avoir des informations sur une zone. Si sa syntaxe est différente les résultats se ressemblent.

Donnez des explications sur les réponses obtenues et leurs différences avec les commandes suivantes :

```
dig rt410gw6a.gtr.tp.
dig @centrex rt410gw6a.gtr.tp.
dig @centrex
dig gtr.tp.
dig gtr.tp. ANY
```

## II. Serveur de noms

Nous allons définir une zone de noms que nous gérerons sur votre machine virtuelle. Nous allons utiliser le serveur de nom `named` distribué sous le nom `bind` ( *Berkley Internet Name Domain* ) La configuration d'une zone peut être assez fastidieuse et je vais laisser la description des détails au manuel (commande `man`). Il faut que le paquet `bind9`[1] soit installé, ce qui ne devrait pas être le cas si vous avez utilisé le bon disque `vdi`.

Installer les paquets `bind9` et `bind9-utils`

### *1. Les fichiers de configuration existants*

Je vous propose comme d'habitude un parcours rapide des fichiers de configuration.

Le principal est `/etc/bind/named.conf` que nous ne modifierons pas, il ne fait

1 Si le paquet bind9 n'est pas installé, lancer les commandes
```
# apt-get update ; apt-get install bind9 bind9-utils
```

que les inclusions des fichiers named.conf.options, named.conf.local et named.conf.default-zones.

### a) Le fichier named.conf.default-zones

Il contient un certain nombre de zones en tant que master. Il s'agit des zones par défaut qui permettent le fonctionnement interne de la machine. Une description rapide est donnée dans les fichiers. Ce sont des zones dont la résolution doit être effectuée en local. Nous allons les examiner de plus près.

- « localhost » et « 127.in-addr.arpa » sont les zones directe et inverse pour l'interface de loopback. Ouvrir le fichier db.local
  - On voit des commentaires (;) puis
    la directive $TTL 604800 qui signifie que toutes les valeurs ont une durée de vie par défaut de 7 jours. Les TTL ne sont donc pas présentes dans le reste du texte
  - On a une ligne de type SOA avec les différents paramètres déjà vus au paragraphe I. 2. b) page 4.
  - On voit à plusieurs endroits un caractère « @ » : il signifie l'origine (directive $ORIGIN) c'est une troisième façon de dénommer le domaine ou la zone. À défaut de placer la directive $ORIGIN dans le fichier de zone c'est la zone indiquée dans le fichier named.conf* qui est valide. Ici « @ » signifie donc « localhost ».
  - On a ensuite
    - une définition du serveur de nom (type NS)
    - une adresse IPv4
    - une adresse IPv6
- La zone suivante est « 127.in-addr.arpa » définie dans le fichier db.127. Le nom des fichiers n'a aucune importance en dehors de la clarté pour celui qui doit maintenir ses zones. Le nom de la zone est par contre très significatif.
  - Dans le fichier db.127 les mêmes lignes : $TTL, type SOA et NS mais
  - la dernière ligne est étrange :

```
1.0.0   IN      PTR     localhost.
```

On a vu que si une dénomination ne se terminait pas par un « . » elle était complétée par le nom de la zone. Cela donnerait si l'on avait tout écrit sans utiliser la technique de complétion

```
1.0.0.127.in-addr.arpa.   IN      PTR     localhost.
```

Cela s'interprète donc comme « dans le domaine des adresses Internet, 127.0.0.1 pointe vers (*doit être compris comme*) localhost. ».

L'inversion des octets permet la complétion automatique par le nom de la zone et va donc nous éviter un fastidieux remplissage de textes toujours identiques.

Vérifiez avec la commande suivante

`host 127.0.0.1`

**b) Les autres fichiers de configuration**

Nos définitions particulières seront mises dans le fichier `named.conf.local`.

Voir `man named.conf` pour l'explication complète des différents champs.

Nous utiliserons le fichier `named.conf.options` pour modifier le comportement du serveur.

```
$TTL 3h
;
; gtr.tp
;

@       IN      SOA     maat.unice.fr. admin.iutsoph.unice.fr.  (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum
              IN      NS      maat.unice.fr.
              IN      MX      10  iutsoph.unice.fr.


gtr408lp1    IN A 10.4.108.91

gtr408p0     IN A 10.4.108.100
             IN MX 10 iutsoph.unice.fr.

gtr408p1     IN A 10.4.108.101
             IN MX 10 iutsoph.unice.fr.
www1a        IN CNAME gtr408p1
www1b        IN CNAME gtr408p1
www1c        IN CNAME gtr408p1

gtr408p2     IN A 10.4.108.102
             IN MX 10 iutsoph.unice.fr.
www2a        IN CNAME gtr408p2
www2b        IN CNAME gtr408p2
www2c        IN CNAME gtr408p2

; fichier tronqué...
```

*Listing 2: Exemple de fichier de zone directe*

## *2. C'est ma zone !*

Tout d'abord une remarque importante : notre serveur DNS ne va pas décrire une véritable zone, ni de véritables machines avec de véritables adresses IP. Au contraire !

#### a) Choix des adresses et du nom de zone

Il est fortement conseillé de choisir une zone d'adresses distribuées qui ne soit pas en interférence avec des zones qui existent localement : je conseille la zone d'adresses 192.168.({5,7,9}0+**Pc**).0 où Pc est le numéro du Pc sur le quel vous travaillez. Le troisième octet est calculé en ajoutant 50 en salle 410, 70 en salle 405 et 90 en salle 102 pour éviter la collision de noms et d'adresses avec le sous-réseau 192.168.P.0/24 utilisé dans d'autres TP. Ce numéro de Pc correspond à la fin de l'adresse de votre Pc physique.

Même chose pour la zone : vous prenez comme nom de zone « dom{Pc}.tp », ce qui donne sur le Pc n°7, un domaine « `dom7.tp` ».

> Avant de commencer à remplir les fichiers de zone, définissez ce que vous devez avoir comme noms de machines et adresses IP. Les noms de fichiers doivent être choisis de manière explicite pour que la maintenance soit facile.

**Autres conseils** : le `bind` est très peu permissif. Il ne doit y avoir aucune erreur de syntaxe pour que cela fonctionne, en particulier il veut des « ; » à toutes les fins de ligne. Il est donc intéressant de *débugger* le fichier de zone avec la zone directe d'abord et, seulement ensuite, établir la zone inverse. Je vous rappelle que les messages d'erreur peuvent être vus en interactif par la commande :

```
tail -f  /var/log/syslog
```

### *b) La zone directe*

> Commencez par remplir le fichier de zone directe en vous inspirant de ce que nous avons vu dans le fichier `db.local` et du listing 2 page 8 et les contraintes indiquées au paragraphe précédent.
>
> Vous devez avoir des types SOA, NS, A, MX et CNAME. Il vous faut adapter certains champs avec les noms et adresses qui correspondent à VOTRE machine virtuelle en tant que serveur de noms.
>
> Dans le fichier `named.conf.local` déclarer la zone en vous inspirant de `named.conf.default-zones`. Il faut vous déclarer comme « master » et désigner le fichier de zone que vous avez rempli.
>
> Il faut aussi enlever le contrôle d'identité que nous ne gérons pas encore : commentez par un # la ligne DNSSEC du fichier `named.conf.options`.
>
> Redémarrer le service `bind`
>
> ```
> systemctl restart named
> ```
>
> En interrogeant votre serveur vous devriez obtenir les adresses que vous avez déclarées. Pas sûr, utilisez l'option `-v` pour `host` et voyez qui vous répond ?

### c) Changement d'adresse !

Votre nouveau serveur est en DHCP et c'est le serveur DHCP qui décide pour vous qui est votre serveur de noms.

Il faut modifier votre fichier `interfaces` pour passer votre interface `en1` en statique en conservant la même adresse IP. Comme serveur de noms indiquez `localhost` et comme zone de recherche ajoutez la zone que vous venez de créer. Vous pouvez vous inspirer du site `http://centre.gtr.tp/dokuwiki/doku.php?id=sys:lin:interfaces` .

Avec `ifdown en1` et `ifup en1` vous devriez améliorer le fonctionnement.

Si ce n'est pas bon examinez le `syslog` !

```
$TTL 3h
;
;
;
@       IN      SOA     maat.unice.fr. admin.iutsoph.unice.fr.  (
                                        1997022700 ; Serial
                                        28800      ; Refresh
                                        14400      ; Retry
                                        3600000    ; Expire
                                        86400 )    ; Minimum
              IN      NS      maat.unice.fr.

91        IN      PTR     gtr408lp1.gtr.tp.
100       IN      PTR     gtr408p0.gtr.tp.
101       IN      PTR     gtr408p1.gtr.tp.
102       IN      PTR     gtr408p2.gtr.tp.
103       IN      PTR     gtr408p3.gtr.tp.
104       IN      PTR     gtr408p4.gtr.tp.
; fichier tronqué...
```

*Listing 3: Exemple de fichier de zone inverse*

### d) Zone inverse

Quand la zone directe fonctionne, passez à la zone inverse. Même chose d'abord le fichier de zone (attention aux « ; » et aux « . »), puis déclarez la zone dans `named.conf.local`. Aidez-vous des exemples du listing n° 3 page 10.

Testez.

## 3. Et le reste du monde

Le service de noms ne fonctionne plus que pour la zone déclarée localement. Le tester (commandes `host` et `dig`).

Déclarer 134.59.136.1 comme `forwarders` dans le fichier de configuration ( `/etc/bind/named.conf.options` ).

Tester la résolution de noms dans les 2 sens et pour des noms non locaux.

## 4. Nouvelle génération

Et si on veut en fabriquer par dizaines, faut-il rentrer à la main ?

Non, le bind y veille : on peut définir beaucoup d'adresses en une seule ligne !

Imaginez que vous devez déclarer dans votre serveur DNS dans la zone que vous venez de créer une liste de noms dépendants d'un nombre :

- Vous auriez à déclarer des machines toto129 à toto229 d'adresses 192.168.[Pc+{50,70,90}].129 à 229[1]. Une solution simple : vous tapez les 100 lignes ! Mais je vous ai dit, le Bind y veille !

Dans votre fichier de zone directe vous mettez :

```
$generate 129-229/1 toto$ A 192.168.[Pc+{50,70,90}].$
```

Si j'explique : on génère dans les bases du DNS (pas dans le fichier de configuration) de 129 à 229 les 100 lignes où le caractère $ est replacé itérativement par des nombres de 129 à 229. En particulier nous définissons ainsi l'adresse de toto200.dom[Pc].tp. . Il peut être de bon aloi de remplacer la chaine « toto » par quelque-chose de plus approprié.

```
zone "gtr.tp" {
type slave ;
file "bak.db.gtr.tp";
masters { 134.59.136.1 ; };
};

zone "110.4.10.in-addr.arpa" {
type slave ;
file "bak.db.10.4.110";
masters { 134.59.136.1 ; };
};

zone "105.4.10.in-addr.arpa" {
type slave ;
file "bak.db.10.4.105";
masters { 134.59.136.1 ; };
};
```

*Listing 4: Zones à ajouter pour le serveur secondaire*

## 5. Du primaire au secondaire

Les zones que vous venez de créer sont toutes de types « master » mais il est intéressant d'avoir, par exemple, sur Sophia, une copie du serveur de noms qui pour l'ensemble de l'Université est à Valrose à Nice. En cas de rupture du réseau, de panne de la machine maître, il en faut d'autres qui prennent le relai temporairement mais aussi qui sont plus proches. On les appelle serveurs secondaires, *slave* en anglais.

1 La formule un peu compliquée signifie par exemple que pour le pc n°7 dans la salle 102, nous aurons les adresses 192.168.97.129 à 229. dans la salle 410 cela aurait 192.168.57… comme expliqué au paragraphe n° II. 2. a) page 9.

On va rendre votre machine virtuelle secondaire de la machine `maat.unice.fr` pour la zone « `gtr.tp` » et les zones inverses « `110.4.10.in-addr.arpa` » et « `105.4.10.in-addr.arpa` ». Pour cela (si le serveur l'autorise, ce qui est le cas) il suffit de copier dans le fichier `named.conf.local` les 3 zones du listing 4 page 11.

Faire redémarrer le bind9, vérifiez le fonctionnement et les messages dans `syslog`

Jeter un coup d'œil dans le répertoire `/var/cache/bind/`

Fichier *TP-4_DNS-2223.odt* imprimé le 22 mars 2023