Installation d'un serveur DNS (Domaine name System) sous Ubuntu Server 12.10

Table des matières

- **Un peu d'histoire**

L'ancêtre d'Internet est le réseau Arpanet - un réseau d'ordinateurs créé pour permettre aux chercheurs, aux Etats Unis, de consulter et échanger leurs travaux et partager leurs ressources.

C'est en 1969 que R. Taylor, de l'agence ARPA, contre l'avis des fabricants d'ordinateurs, relie 4 ordinateurs de quatre universités différentes, entre eux.

En 1973, le réseau compte 35 ordinateurs mais déjà, depuis l'année précédente, apparaît la nécessité d'utiliser un protocole commun de communication, non entre ordinateurs d'un même réseau mais entre réseaux ayant, chacun, leur langage.

Dès 1974, V. Cerf et B. Kahn publient TCP/IP (Transfert Control Protocol/Internet Protocol). TCP/IP sera totalement libre et dans le domaine public. Immédiatement, d'autres réseaux sont créés et utilisent TCP/IP pour communiquer entre eux dont CSNET en 1979 et NFSNET en 1980. Le réseau des réseaux finira par prendre le nom d'Internet.

En ces années du Neandertal des réseaux, Arpanet, toute l'information dont a besoin un ordinateur pour identifier les autres ordinateurs (hosts) du réseau tient dans un simple fichier texte appelé hosts.txt. C'est une bête liste de correspondance entre « Nom d'un ordinateur » et « Adresse de l'ordinateur » car il est plus aisé pour l'humain de dire "Je me connecte sur la machine de telle université ou tel laboratoire..." plutôt que de dire "Je me connecte sur la machine 063.124.231.077".

Le travail de maintenance de cette liste est confié au Network Information Center (« NIC ») du « Stanford Research Institute (« SRI ») de Menlo Park en Californie. Les modifications à cette liste sont soumises au NIC qui les compile en un nouveau hosts.txt, une ou deux fois par semaine. Un mécanisme de « résolution de noms » est implanté dans tous les systèmes d'exploitation des ordinateurs, quelque soit leur fabriquant.

DNS remplace Hosts en 1983

Les ordinateurs, qui n'étaient que quelques dizaines, se multiplient et se mettent en réseau. Rapidement, le travail de maintenance de cette liste et la charge du serveur du « Stanford Research Institute (« SRI ») deviennent énormes. Les mises à jour deviennent si fréquentes que de nouvelles mises à jour sont introduites avant même que les mises à jour précédentes ne soient déployées sur le réseau Arpanet. En sus, il n'y a aucune autorité de gouvernance ni de convention de nommage des ordinateurs et deux ordinateurs peuvent prétendre porter le même nom ce qui fait s'écrouler tout le système.

Puis se sont d'autres réseaux, concurrents d'Arpanet, qui voient le jour. Et puis ces réseaux, ces "Networks" en anglais, s'interconnectent entre eux. C'est la naissance de l'Internet, le réseau des réseaux.

Paul Mockapetris, de l'University of Southern California (« USC »), propose l'architecture Domain Name System (« DNS ») en 1983 pour résoudre ces problèmes. Entre autre, il propose le nommage hiérarchisé (les "points" qui séparent des "hiérarchies" dans les noms),

une gestion « locale » de ces hiérarchies par des organismes ayant une délégation d'autorité sur une zone géographique, assurant ainsi une mise à jour simplifiée et l'unicité des noms. Il propose enfin un mécanisme simple de mise à disposition pour l'ensemble de l'Internet.

Le DNS (Domain Name System) repose sur une constellation d'ordinateurs appelés "serveurs de noms de domaine", tous redondant et tous propageant leurs mises à jour locales vers tous les autres, en permanence. Le maillage du réseau et la redondance de ses nœuds est telle qu'une attaque contre plusieurs serveurs de noms de domaines les faisant chuter en même temps n'entamerait pas le fonctionnement de l'Internet mondial.

Présentation rapide d'un système DNS

L'architecture de réseau TCP/IP sur lequel est basé Internet et la plupart des réseaux locaux actuels, utilisent des adresses IP numériques du type 192.168.0.1. Mais pour faciliter la lecture de ces adresses par l'homme, un système permet de transformer ces adresses en adresses plus lisibles comme www.coagul.org

Pour effectuer cette opération, il est nécessaire d'utiliser des serveurs DNS. Un serveur DNS fera donc la correspondance entre les adresses IP et les noms des domaines.

Un serveur DNS s'occupe en général d'un domaine limité et s'occupe de transmettre les questions à d'autres serveurs s'il ne connaît pas la réponse.

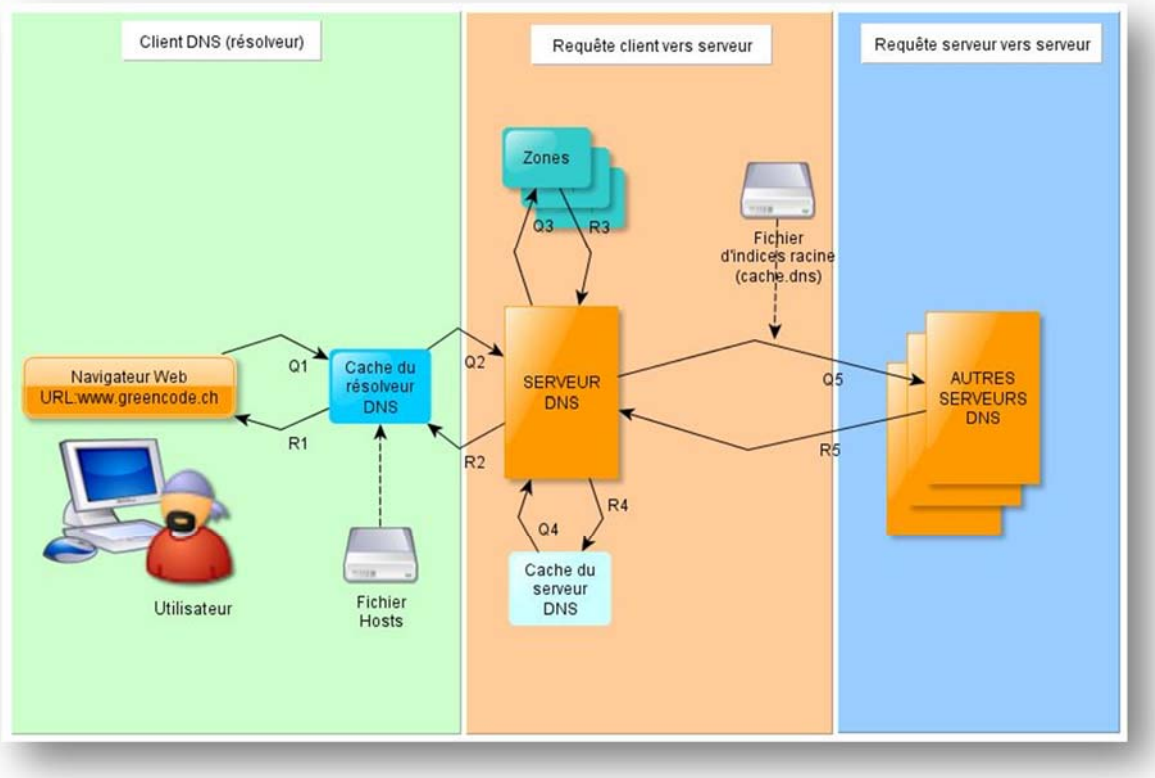
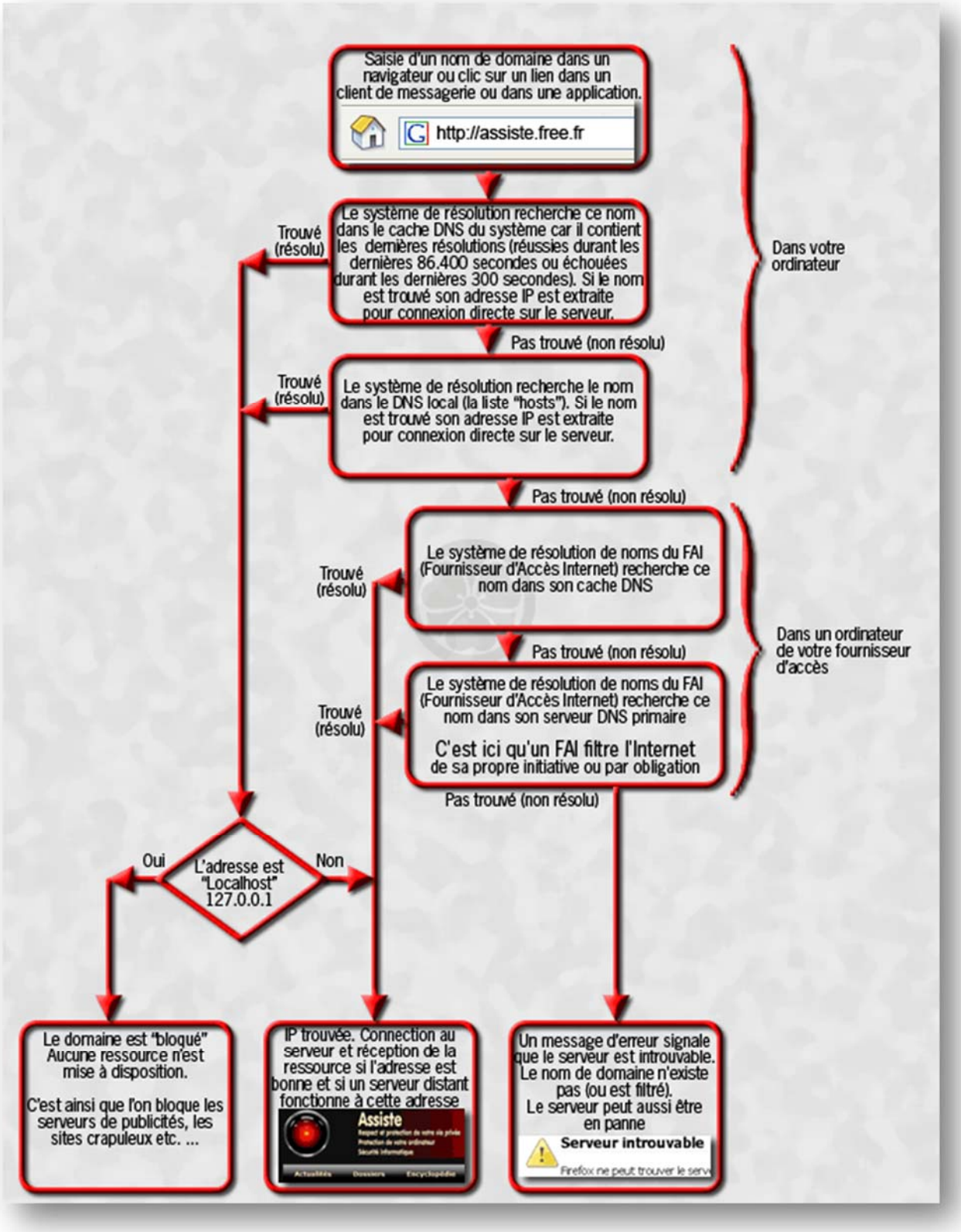


Schéma de principe de la résolution des noms de domaine



- **Principe de fonctionnement de la recherche de noms**

Lorsque qu'une demande de résolution de nom est demandée, le système commence par regarder le fichier « **/etc/hosts.conf** :

```
############################
order hosts,bind
multi on
############################
```

La première ligne de ce fichier indique qu'il faut commencer la recherche en regardant la table hosts locale et ensuite il faut interroger le serveur DNS.

La table hosts locale est enregistrée dans le fichier « **/etc/hosts** » elle contient une table de correspondance entre des adresses IP et des noms, elle ressemble à :

```
#################################################################
127.0.0.1       localhost.localdomain   localhost
172.25.205.250  Ubuntu-SERVER-64.ghost.fr  Ubuntu-SERVER-64
#################################################################
```

La première ligne est obligatoire pour que le système fonctionne même quand le réseau est désactivé. L'adresse IP 127.0.0.1 est toujours associée au nom **localhost.**

Les lignes suivantes peuvent être ajoutées manuellement pour faire la correspondance entre des adresses IP et des noms. C'est ce qui est fait en l'absence de serveur DNS.

Si le résultat n'est pas trouvé dans la table **hosts**,le système recherche le serveur DNS indiqué dans le fichier « **/etc/resolv.conf** » :

```
################################################
domain    ghost.fr
search ghost.fr          # Votre domaine
nameserver 172.25.205.250     # Votre DNS
nameserver 194.168.1.254      # DNS de votre FAI
################################################
```

La première ligne indique quel domaine il faut ajouter aux noms si celui-ci n'est pas indiqué lors d'une demande de résolution de nom.

Exemple :

- ping monserveur.mondomaine.com -> Aucun domaine ne sera ajouté lors de la résolution du nom, car le domaine est fourni.

- ping monserveur -> Le domaine mondomaine.com, sera ajouté avant d'effectuer la demande de résolution du nom (La recherche du nom, portera donc sur monserveur.mondomaine.com)

La deuxième ligne indique le serveur DNS principal.

Et c'est donc le serveur DNS qui sera chargé de donner le résultat s'il connaît la réponse ou de transmettre la question à un autre serveur DNS.

Si le serveur principal n'est pas disponible, le serveur DNS indiqué sur la ligne suivante sera utilisé.

Pourquoi installer un serveur DNS

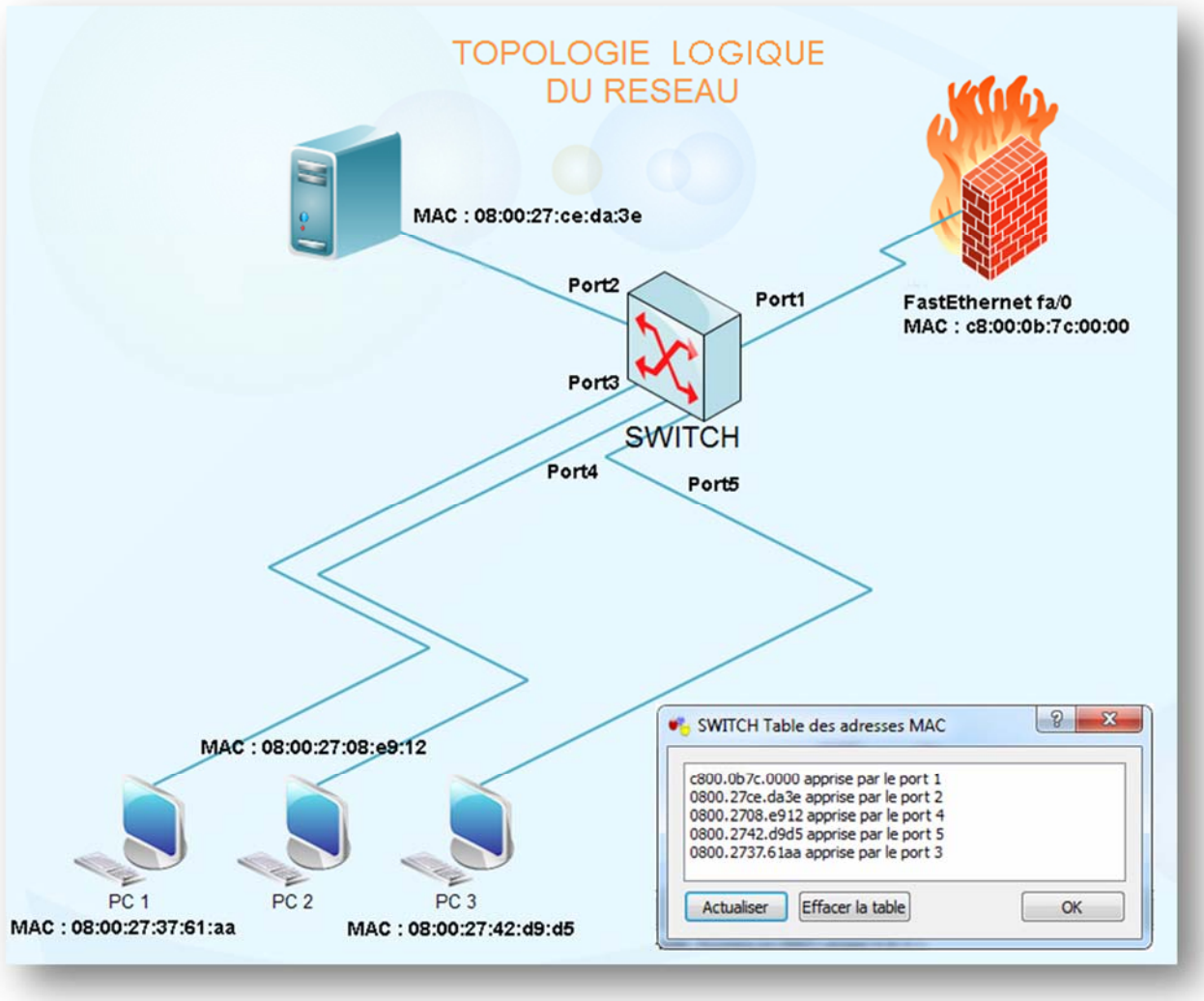
Pour au moins deux raisons :

- Éviter de tenir à jour la table hosts de chaque poste client d'un réseau.

- Avoir un cache DNS qui accélère la recherche des noms.

- Sur un réseau local, un serveur DNS permet d'accélérer le trafic sur le réseau car de nombreux services ont besoins d'un serveur DNS bien configuré pour fonctionner correctement (WEB, POP, SMTP,..)

Installation du serveur de DNS Bind9 et configuration



- **Installation de Bind 9**

L'installation est identique sur sur Debian ou Ubuntu, il faut installer le paquet suivant :

Sous Debian, il faut installer le paquet suivant :

```
#########################################################
 aptitude install bind9 ou apt-get install bind9
#########################################################
```

La configuration sera différente sur Ubuntu Serveur 12.10 , car certaines restrictions de sécurité empêchent le service DNS de fonctionner correctement.

Ces configurations spéciales seront vues en détails à la fin de cette section.

Configuration du serveur de noms (DNS)

Préparation des Fichiers système

Les fichiers systèmes suivants se trouvent dans les répertoires « **/etc** » et « **/etc/network** »

- **/etc/resolv.conf**
- **/etc/hosts**
- **/etc/network/interfaces**
- **/etc/hostname**

Adressage IPV4 : Editer le fichier **« /etc/network/interfaces »**

Ce fichier décrit la configuration des interfaces réseau disponibles sur votre OS**.**

La configuration de la carte principale dépendra du nombre de cartes présentes sur votre système, en générale une seule carte est présente qui est nommée **eth0.**
Sur les exemples suivant la carte principale est l'interface **eth1**

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet dhcp

auto eth1
iface eth1 inet static
address 172.25.205.250
        netmask 255.255.255.0
        gateway 192.168.0.254
        dns-nameservers 192.168.0.254
```

- Cette adresse est la boucle locale du système
- Vous devez remplacer l'attribution de l'adresse IP en « static » pour votre interface
- Cette adresse est celle qui définira votre serveur
- Cette adresse est le masque de sous réseau
- Cette adresse est la passerelle par défaut
- Celle adresse est le serveur DNS qui vous sert normalement à résoudre les adresses

Editer ensuite le fichier « **/etc/resolv.conf** »

```
domain  ghost.fr
search  ghost.fr
nameserver      172.25.205.250
nameserver      192.168.0.254
nameserver      8.8.8.8


"/etc/resolv.conf" [lecture-seule] 5L, 102C
```

Dans ce fichier vous devez y répertorier :

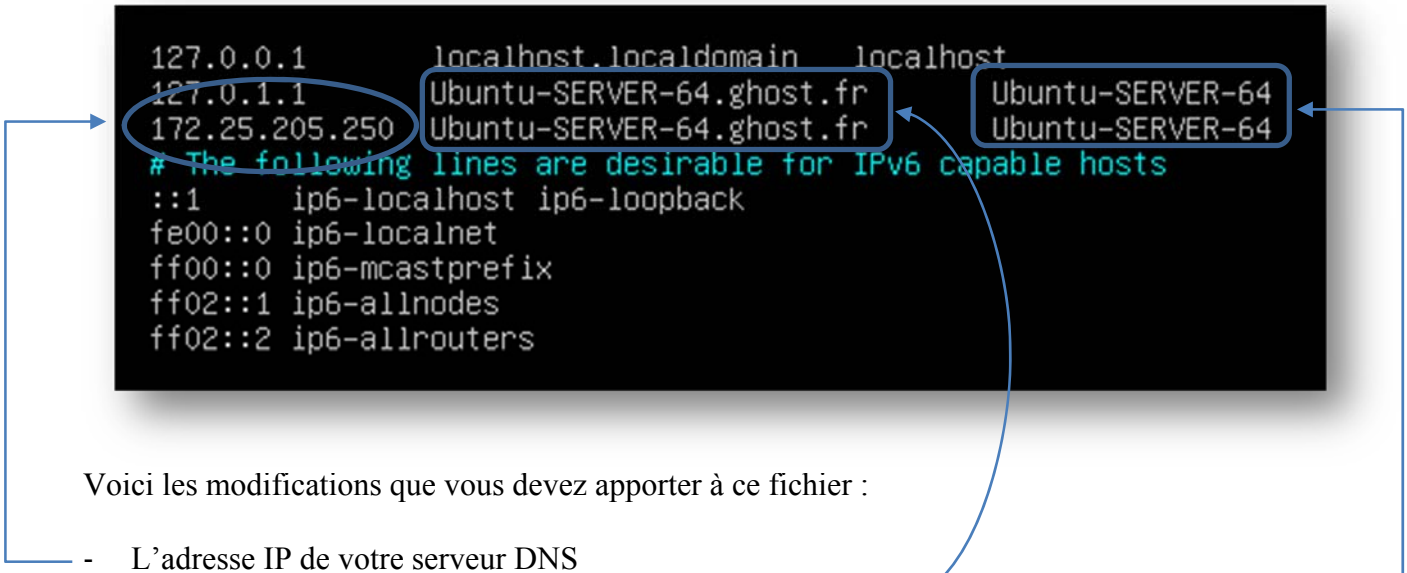
- Votre nom de domaine
- Le domaine sur lequel seront exécutées les recherches
- Et les serveurs DNS avec en premier prioritairement votre serveur DNS qui correspond à l'adresse IP STATIC que vous avez fixé sur votre interface principale plus haut dans la configuration di fichier **« /etc/network/interfaces »**

Le nom de votre ordinateur servira dans tous les paramétrages du serveur DNS , vous devez donc choisir un nom qui définira au mieux votre système.

Le nom de votre ordinateur est enregistré dans le fichier **« /etc/hostname »**
Si vous désirez changer le nom de celui-ci effectuez les 2 lignes de commande suivantes.

```
########################################
echo server.example.com > /etc/hostname
/etc/init.d/hostname restart
########################################
```

Modification du fichier : **« /etc/hosts »**



Voici les modifications que vous devez apporter à ce fichier :

- L'adresse IP de votre serveur DNS
- VOTREhostname.VOTREdomaine.TLD (Top-level domain :com,.fr,.net….)
- VOTREhostname
- La première ligne ne doit pas être modifiée puisque c'est la boucle locale.

Configuration de BIND9

Les fichiers de configuration de **Bind9** sont dans le répertoire : **« /etc/bind »**

- ***/etc/bind/named.conf***
- ***/etc/bind/named.conf.options***
- ***/etc/bind/named.conf.default-zones***
- ***/etc/bind/named.conf.local***
- ***/etc/bind/named.conf.log***
- ***/etc/bind/db.mondomaine (FQDN)***
- ***/etc/bind/db.mondaine.inv***

La configuration du serveur se fera en modifiant ou en créant les fichiers suivants :

→ Fichier de Configuration Principal (/etc/bind/named.conf)

Le fichier de Configuration principal « **/etc/bind/named.conf** » contient la liste des zones (ou domaines) que le serveur DNS doit prendre en charge.

```
// This is the primary configuration file for the BIND DNS server named.
//
// Please read /usr/share/doc/bind9/README.Debian.gz for information on the
// structure of BIND configuration files in Debian, *BEFORE* you customize
// this configuration file.
//
// If you are just adding zones, please do that in /etc/bind/named.conf.local

include "/etc/bind/named.conf.options";
include "/etc/bind/named.conf.local";
include "/etc/bind/named.conf.default-zones";
```

Pour tous les domaines à résoudre sur votre serveur primaire, il est nécessaire de rajouter les directives indiquant les fichiers de configuration à utiliser pour chacun de vos domaines.Aucune modification n'est à apporter à ce fichier, les configurations étant faites dans les fichiers **named.conf.options** et **named.conf.default-zones**

Voici un exemple de description de zone inclue dans le fichier **« /etc/bind/named.conf.default-zones »** que nous verrons plus tard :

```
zone "mondomaine.com" {
        type master;
        file "/etc/bind/db.mondomaine.com";
        forwarders{};
};
```

mondomaine.com : Nom du domaine à prendre en charge

type master : Cette ligne indique que le serveur est le serveur principal de ce domaine.

file "/etc/bind/db.mondomaine.com" : Cette ligne donne le chemin du fichier qui contiendra la correspondance entre les noms et les adresses IP pour ce domaine.

→ Fichier de configuration /etc/bind/named.conf.options

Ce fichier définit un serveur DNS stable en cas de problèmes avec votre propre serveur.Il contient également toute une série d'options dont voici les détails :

```
include "/etc/bind/named.conf.log";


options {
        directory "/var/cache/bind";

        // If there is a firewall between you and nameservers you want
        // to talk to, you may need to fix the firewall to allow multiple
        // ports to talk.  See http://www.kb.cert.org/vuls/id/800113

        // If your ISP provided one or more IP addresses for stable
        // nameservers, you probably want to use them as forwarders.
        // Uncomment the following block, and insert the addresses replacing
        // the all-0's placeholder.

        forwarders {
                172.25.205.250; 192.168.0.254;
        };
        listen-on-v6 {
        internal
        };
        allow-query {
        internals;
        };
        allow-recursion {
        internals;
        };

        version none;

        auth-nxdomain no;    # conform to RFC1035
};
```

- Pointe vers le fichier de log que nous configurerons plus loin

- L'option "**forwarders**" permet de rediriger les requêtes qui ne sont pas résolues par notre serveur vers un serveur DNS distant (serveurs DNS de votre FAI par exemple).
  Cela permet d'utiliser le cache d'un serveur déjà existant et donc d'obtenir des temps d'accès plus rapides. Si la requête DNS n'est pas résolue par le serveur DNS "*distant*" alors la requête sera envoyée aux serveurs DNS racine.
- **allow-recursion :** la configuration de Bind par défaut comporte une "*faille*" de sécurité, en effet la configuration autorise des tierces personnes à utiliser le serveur DNS (*sans demander la permission*). Cela fait de Bind un serveur DNS relais !! Pour corriger cette faille nous allons ajouter une option dans le fichier, Ce qui a pour incidence que l'utilisation de Bind **ne sera autorisée que sur le serveur même**.
- L'option "**version**" permet de dissimuler la version de Bind, en effet une personne malveillante peut vouloir récupérer la version de votre Bind afin de mener une attaque contre ce dernier si il n'est pas à jour. Vous pouvez soit mettre non ou bien la syntaxe que vous voulez ex : **version "ghost server"**

→ Fichier de configuration /etc/bind/named.conf.default-zones

Ce fichier contient la configuration de la boucle locale et de la boucle locale inverse.

Vous déclarerez, dans ce fichier, les zones (sous domaines) de votre domaine

```
// prime the server with knowledge of the root servers
zone "." {
        type hint;
        file "/etc/bind/db.root";
};

// be authoritative for the localhost forward and reverse zones, and for
// broadcast zones as per RFC 1912


zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};
```

→ Fichier de configuration /etc/bind/named.conf.local

Comme vu dans le fichier **/etc/bind/named.conf.options,** en face de certaines options on peut trouver en valeur : **internals**.
Cette valeur correspond aux Access Control List (**acl**).
Si on les déclare en internals il faut préciser à bind9 quelles sont les @IP autorisées à communiquer avec bind.
Ce fichier contient également la configuration du serveur root local et son inverse.
Pour cela il faut éditer le fichier **/etc/bind/named.conf.local** comme suit :

```
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";
        acl internals { 127.0.0.0/8; 172.25.205.0/24; };

zone "ghost.fr" {
        type master;
        file "/etc/bind/db.ghost.fr";
        allow-update { key rndc-key; };
};

zone "205.25.172.in-addr.arpa" {
        type master;
        file "/etc/bind/db.ghost.fr.inv";
        allow-update { key rndc-key; };
};
```

Seront définies ici les @IP des réseaux internes autorisées à se connecter au DNS. Il y a bien sûr la boucle locale 127.0.0.0/8 et le réseau à déclarer 172.25.205.0/24 .

- votre domaine
- le nom de votre fichier de zone « db.votre domaine »
- la partie réseau (sans l'hôte) inversée de votre serveur DNS
- le nom du fichier de zone inverse « db.votre domaine.inv»

→ Fichier de configuration /etc/bind/named.conf.log

C'est le fichier de configuration des logs de **bind9**.
Par défaut, il n'existe pas, il faudra donc le créer et y copier les informations suivantes de configuration des différents fichiers de log :

```
logging {
        channel update_debug {
                file "/var/log/named/update_debug.log"
versions 3 size 100k;
                severity debug;
                print-severity  yes;
                print-time      yes;
        };
        channel security_info {
                file "/var/log/named/security_info.log"
versions 1 size 100k;
                severity info;
                print-severity  yes;
                print-time      yes;
        };
        channel bind_log {
                file "/var/log/named/bind.log" versions 3 size
1m;
                severity info;
                print-category  yes;
                print-severity  yes;
                print-time      yes;
        };

        category default { bind_log; };
        category lame-servers { null; };
        category update { update_debug; };
        category update-security { update_debug; };
        category security { security_info; };
};
```

Il faut maintenant créer tous les fichiers de log à utiliser :

```
touch   /var/log/named/update_debug.log
touch   /var/log/named/security_info.log
touch   /var/log/named/bind.log
chown bind:bind   /var/log/named/update_debug.log
chown bind:bind   /var/log/named/security_info.log
chown bind:bind   /var/log/named/bind.log
```

→ Fichier de configuration de zone /etc/bind/db.mondomaine (FQDN)

Le fichier de zones réunit tous les enregistrements qui indiquent vers quels serveurs pointent votre domaine et ses sous-domaines.

Vous devez créer votre fichier de zones afin de définir quelles adresses et machines devra utiliser votre serveur DNS.

Voici comment doit être renseigné ce fichier

```
$ORIGIN mondomaine.TLD.
$TTL  1600
@ IN SOA myhostname.mondomaine.TLD.   @mail_de_l'admin_du_serveur. (
                  2012122210 ; Serial
                   604800          ; Refresh
                    86400          ; Retry
                  2419200          ; Expire
                   604800 )  ; Negative Cache TTL

@                 IN    NS    myhostname.mondomaine.TLD.
@                 IN    MX    10    smtp.mondomaine.TLD.
myhostname        IN    A     @IPV4_du_serveur_bind
smtp              IN    A     @IPV4_du_serveur_smtp
www               IN     A    @IPV4_du_serveur_www
webmail           IN    CNAME smtp
```

Attention à la syntaxe, il faut parfois ajouter un '.' à la fin du FQDN (mon domaine) et remplacer le '@' par un point dans l'@mail de l'administrateur du serveur.

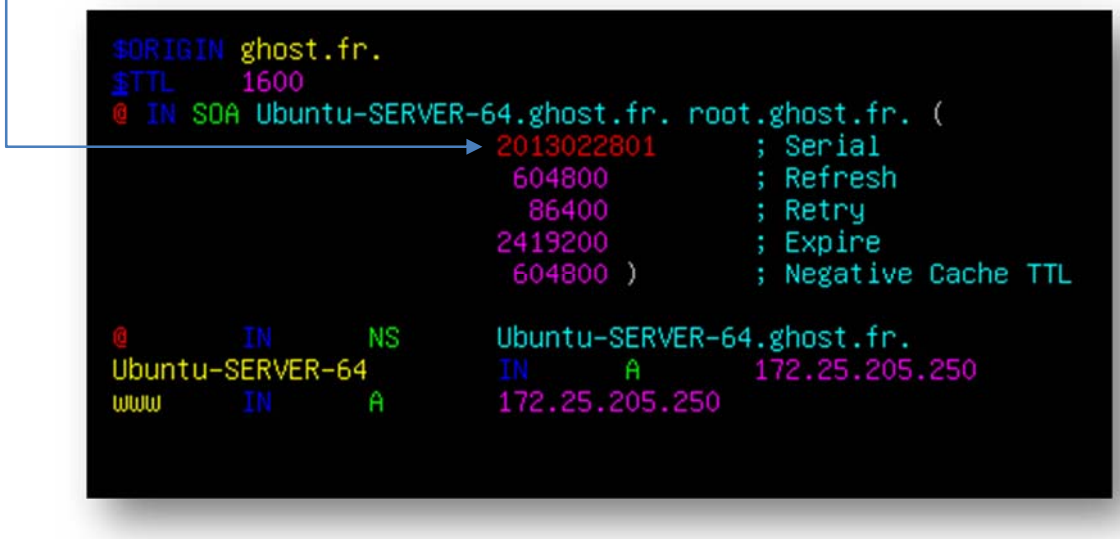
Dans la deuxième partie du fichier, on trouve les enregistrements des différents serveurs reliés à bind.

@	IN	NS		==> Définit le nom complet du serveur Bind
@	IN	MX	10	==> Enregistrement du champ MX du serveur smtp
smtp	IN	A		==> Enregistrement du nom de machine smtp
webmail	IN	CNAME		==> Enregistrement alias du webmail (sur smtp)
www	IN	A		==> Enregistrement de l'adresse du serveur web

Note : smtp, www et myhostname sont les noms de machine des serveurs reliés à Bind. Grâce à ces enregistrements, on a accès à ces serveurs par leur nom. Plus besoin de taper les @IP.

→ N° de série à incrémenter à chaque modification de ce fichier. Ce N° est utilisé par les serveurs esclaves pour lui indiquer qu'il doit mettre à jour sa base. Par commodité ce n° est une date à l'envers.

Voici un exemple de fichier de zone sur le domaine ghost.fr le nom du fichier est donc : **db.ghost.fr**



Si vous n'avez pas de serveur web ou de messagerie, les champs « smtp » et « www » ne devront pas être complétés.

En détail :

Bind9 nous permet de définir un nombre très limité de variables, qui seront placées au début du fichier et établiront des valeurs par défaut pour les enregistrements. Il s'agit de :

- **$TTL**, permet de définir, en secondes et dans un intervalle qui va de 0 à 2147483647, soit, si je ne me trompe pas dans mes calculs, près d'une dizaine d'années, le délai maximum pendant lequel un enregistrement pourra être gardé en cache. Avec 86400, le cache sera vidé, et les fichiers relus, toutes les 24 heures.
- **$ORIGIN** déclare un nom de domaine à ajouter pour reconstituer le nom pleinement qualifié
- **$INCLUDE** permet d'indiquer le chemin d'accès d'un fichier à utiliser ; après le chemin, on peut indiquer un nom de domaine différent de celui qui a été défini par $ORIGIN. On comprend que ces deux directives servent essentiellement à la gestion de domaines et sous-domaines d'architecture complexe.

Passons donc à l'analyse de notre premier enregistrement, le **SOA**.

On se souvient que notre serveur de noms est maître sur la zone *ghost.fr,* ce pourquoi il a été défini avec le type **master** dans **named.conf.default-zones**. Notre fichier de zone doit donc commencer par un enregistrement dit **SOA** pour ***start or autority***, qui définira le nom du serveur , l'adresse électronique de l'administrateur, et quelques paramètres.

- ensuite, le **TTL**. Comme elle ne diffère pas de celle qui a été définie dans **$TTL**, il est inutile de spécifier une valeur.
- après, le type d'information. Sur un serveur maître, il existe nécessairement un et un seul enregistrement **SOA**, et c'est toujours le premier du fichier de zone.
- pour finir, les données. En l'espèce, elles sont particulièrement fournies puisqu'on trouve :

  - **Ubuntu-server-64.ghost.fr.**, le nom pleinement qualifié du serveur de nom. On ne manquera pas de noter le **.** si discret à droite du **fr** et caractéristique de ces noms absolus. Il permet à Bind de remonter au sommet de l'arborescence des noms.
  - **root.ghost.fr.** est la manière particulière dont Bind prend en compte l'adresse électronique de l'administrateur du serveur, avec un « . » à la place de l'@.
  - ensuite, entre (), un certain nombre de paramètres suivis de commentaires introduits par un « ; », à savoir :
    - d'abord un numéro de série que l'on ne doit pas oublier d'incrémenter à chaque modification du fichier. Ce numéro permet aux serveurs esclaves de savoir s'il y a du nouveau, et de modifier le contenu de leurs bases en conséquence. Il est d'usage de choisir un numéro du type yyyymmdd, suivi d'un numéro d'ordre, mais cela n'est nullement obligatoire, l'important étant que ces numéros soient toujours croissants.
    - **Refresh**, **Retry**, et **Expire** sont des délais, exprimés en secondes, qui vont piloter le comportement des serveurs esclaves. A l'expiration du délai *refresh*, l'esclave va entrer en contact avec le maître ; s'il ne le trouve pas, il essaiera de nouveau à la fin du délai *retry*. Et si, au bout du délai *expire*, il n'est pas parvenu à ses fins, il considérera que le serveur maître a été retiré du service. **minimum**, enfin, détermine, toujours en secondes, la durée de vie minimum du cache. On remarque la disposition des parenthèses, obligatoire pour disposer les informations sur plusieurs lignes.

Passons à la suite, à savoir la résolution du nom du serveur lui-même :

Sur la première ligne, on déclare **Ubuntu-server-64.ghost.fr.** comme étant un serveur de noms.

- Sur la seconde, le nom **Ubuntu-server-64** est associé à une adresse IP, 172.25.205.250. Généralement, les serveurs tournant sur une machine ne servant qu'à ça s'appellent **ns**, lequel ns est aussi arbitraire et traditionnel que le **www** des serveurs web.
- La troisième ligne déclare l'adresse du serveur web **www.**

→ Fichier de configuration de zone inverse /etc/bind/db.mondomaine.inv (FQDN)

Tout le travail effectué jusqu'à présent ne servait qu'à associer une adresse IP à un nom. Pour le bon fonctionnement du réseau, il est nécessaire que l'opération inverse - retrouver un nom à partir d'une adresse IP - soit possible : nombre de protocoles, HTTP et FTP pour n'en citer que deux, vérifient grâce à cette méthode, lors de la connexion d'un client, l'exactitude des informations d'identification qu'il a donné. On utilise pour cela une arborescence dont la racine n'est plus matérialisée par un simple point, mais par un nom : in-addr.arpa, et où les adresses IP sont consignées en sens inverse de l'ordre habituel.
Pour bien comprendre le mécanisme, il faut se rappeler que les points, dans un nom de domaine, sont des séparateurs hiérarchiques : en parcourant le nom de gauche à droite, on va du plus petit au plus grand.

Pour la résolution inverse, on lit des adresses IP, et on inverse l'ordre : 127.0.0.1 devient ainsi 1.0.0.127., adresse que l'on termine par le domaine conventionnel in-addr.arpa.

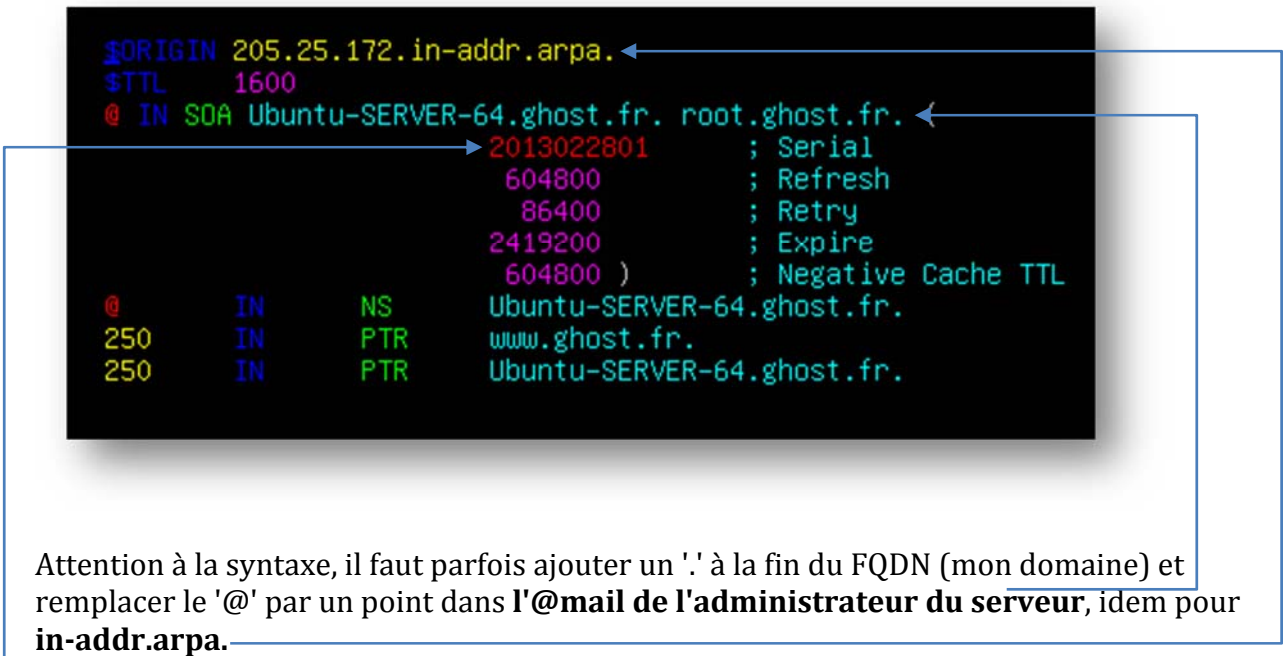
On crée ainsi la zone "0.0.127.in-addr.arpa", qui, dans **named.conf,** renvoie au fichier **named.conf.default-zones**. On remarque que le nom de la zone ne reprend pas le dernier octet de l'adresse IP : celui-ci se retrouve dans le seul enregistrement de n**amed.local .**

On remarque que le nom de la zone ne reprend pas le dernier octet de l'adresse IP : celui-ci se retrouve dans notre fichier **/etc/bind/db.mondomaine.inv** qui contient la correspondance entre la fin de l'adresse IP et le nom du serveur.

Voici comment doit être renseigné ce fichier

```
$ORIGIN partie_réseau_inversée_de_l'@IP_du_serveur.in-addr.arpa.
$TTL  1600
@ IN  SOA   myhostname.mondomaine.TLD. @mail_de_l'admin_du_serveur. (
                2012122210 ; Serial
                 604800         ; Refresh
                  86400         ; Retry
                2419200         ; Expire
                 604800 )  ; Negative Cache TTL
@     IN    NS    myhostname.mondomaine.TLD.
Partie_hôte_@IPV4_du_serveur_www      IN    PTR   www.mondomaine.TLD.
Partie_hôte_@IPV4_du_serveur_smtp     IN    PTR   smtp.mondomaine.TLD.
Partie_hôte_@IPV4_du_serveur_Bind     IN    PTR
     myhostname.mondomaine.TLD.
```

Ce qui ressemblerait à ceci :



Attention à la syntaxe, il faut parfois ajouter un '.' à la fin du FQDN (mon domaine) et remplacer le '@' par un point dans **l'@mail de l'administrateur du serveur**, idem pour **in-addr.arpa.**

➔ À chaque changement du fichier de zone ou du fichier de zone inverse, il faut incrémenter le numéro de sérial et relancer **bind9**.

Partie_hôte_@IPV4_du_serveur correspond à la partie de l'@IP qui indique le sous réseau auquel l'hôte appartient. Par exemple pour une @IP telle que :

192.168.0.1/24 la "Partie_hôte_@IPV4_du_serveur" est 1

Mais pour :

192.168.2.3/16 la " Partie_hôte_@IPV4_du_serveur " est 2.3

Et c'est tout : la partie supérieure reprend exactement la déclaration SOA de ***db.ghost.fr***, mais la zone à laquelle renvoie le @ est maintenant 205.25.172.in-addr.arpa., et les seuls enregistrements appropriés sont des pointeurs, alias PTR, auxquels sont associés les derniers octets des adresses des machines. Remarquons que rien n'interdit d'entrer des adresses complètes, 20.200.168.192.in-addr.arpa., de la même manière que l'on peut éventuellement entrer des noms pleinement qualifiés dans les enregistrements de type A. C'est plus long, plus sûr d'un certain point de vue, mais d'un autre côté on multiplie le risque de fautes de frappe.

Et ATTENTION **Bind9** fonctionne très bien sans. Le problème avec **Bind9**, c'est d'ailleurs qu'il fonctionne presque toujours, du moment que **named.conf** ne comporte pas d'erreur de syntaxe, même quand il n'est pas en mesure de fournir les informations demandées.

Modifications à apporter pour utilisation sous Ubuntu Server

Comme précisé au début de ce dossier, certaines restrictions de sécurité empêchent le service DNS de fonctionner correctement.

En effet le logiciel **« apparmor »** semble bloquer l'exécution du service **named**(bind9).

La solution radicale est de supprimer tour simplement **"apparmor"** et d'utiliser un autre service de protection qui ne pose pas de problème.

```
/etc/init.d/apparmor stop
update-rc.d -f apparmor remove
apt-get remove apparmor apparmor-utils
```

Effectuer ensuite l'installation du logiciel SElinux , concurrent principal d'apparmor

```
sudo apt-get install selinux
```

Nous retrouvons donc un système fonctionnel sous Ubuntu Server 12.x avec les mêmes fonctionnalités de **sécurité** qu'auparavant.

ADDENTUM :

Sur Ubuntu 12.x ,il semblerait que des règles soient mises en place au niveau de ce process, empêchant l'écriture dans le répertoire /var/log.

Le fichier de configuration, pour le process **named** lancé par bind9, se situe à l'emplacement **/etc/apparmor.d/usr.sbin.named.**

Le contenu est le suivant :

```
# vim:syntax=apparmor
# Last Modified: Fri Jun  1 16:43:22 2007
#include <tunables/global>

/usr/sbin/named {
  #include <abstractions/base>
  #include <abstractions/nameservice>

  capability net_bind_service,
  capability setgid,
  capability setuid,
  capability sys_chroot,
  capability sys_resource,

  # /etc/bind should be read-only for bind
  # /var/lib/bind is for dynamically updated zone (and journal)
files.
  # /var/cache/bind is for slave/stub data, since we're not the
origin of it.
  # See /usr/share/doc/bind9/README.Debian.gz
  /etc/bind/** r,
  /var/lib/bind/** rw,
  /var/lib/bind/ rw,
  /var/cache/bind/** rw,
  /var/cache/bind/ rw,

  # gssapi
  /etc/krb5.keytab kr,
  /etc/bind/krb5.keytab kr,

  # ssl
  /etc/ssl/openssl.cnf r,

  # dnscvsutil package
  /var/lib/dnscvsutil/compiled/** rw,

  /proc/net/if_inet6 r,
  /proc/*/net/if_inet6 r,
  /usr/sbin/named mr,
  /var/run/named/named.pid w,
  /var/run/named/session.key w,
  # support for resolvconf
  /var/run/named/named.options r,

  # some people like to put logs in /var/log/named/ instead of
having  # syslog do the heavy lifting.
/var/log/named/** rw,  /var/log/named/ rw,}
```

Cette configuration indique qu'il est possible d'effectuer la lecture et l'écriture de fichier au niveau du répertoire **/var/log/named.**

Tout naturellement, la configuration spécifiera un fichier de trace dans celui-ci.

Il suffirait donc de remplacer dans notre configuration l'adresse et le nom du fichier de log.

A confirmer ……..

Problèmes avec le serveur DHCP

Dans le cas où votre serveur hébergerait dans le même temps un serveur **DHCP** , le serveur **DHCP** aura de grandes chances de réinitialiser votre fichier **/etc/resolv.conf.**

Si vous rencontrez ce problème, voici les modifications à effectuer :

- Reconfigurer votre fichier **/etc/resolv.conf** comme nous l'avons effectué plus haut **sur le serveur et sur les système client.**
- Protéger le fichier **resolv.conf** en lecture seule, ce qui empêchera le service **DHCP** de le réinitialiser à chaque démarrage.

```
cd /etc
sudo chattr +i resolv.conf
```

Pour pouvoir remodifier le fichier **resolv.conf** , il faudra rétablir les droits en écriture

```
sudo chattr –i resolv.conf
```

Pour les clients un autre problème peut apparaitre au niveau du fichier **dhcpd.conf** pour la résolution des noms de domaine.

Il faudra alors effectuer les modifications suivantes dans le fichier **dhcpd.conf** **du client :**

```
option domain-name « votredomaine »
option domaine-name-servers « votre serveur DNS »
```

EX : `option domain-name ghost.fr`

`option domaine-name-servers 172.25.205.250`

Mise en fonction du serveur DNS

Le démon qui fait tourner le service DNS Bind9 est **« named »**

Après chaque modification des fichiers de configuration, il faut redémarrer le démon **named** :

```
Sudo service bind9 restart
```

ATTENTION : Il est vivement conseillé de regarder les logs pour vérifier que le démarrage du démon s'est correctement effectué :

```
tail -30 /var/log/syslog
```

Il faut maintenant vérifier la configuration des fichiers de **bind9** avec la commande **« named-checkconf –z »** (à lancer avec sudo).
Si tout a bien été configuré, vous devriez obtenir ceci ceci :

```
sudo named-checkconf -z

zone mon_domaine/IN: loaded serial 2012122210
zone @IP_serveur_inversée.in-addr.arpa./IN: loaded serial
2013022801
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
```

named -g

Le lancement du démon bind9 **« named** » avec l'option –g permettra de détecter d'éventuelles erreurs.
Si la commande précédente renvoie des erreurs, elle est assez verbeuse et elle vous donnera des indications sur les modifications à apporter (quel fichier, quel ligne, quel problème...).
Une mauvaise configuration de Debian (Partie I) peut compromettre l'intégrité des services de **bind9** !

Netstat

```
netstat netstat -ntulp | grep named
```

- **Tests du fonctionnement de la résolution des noms**

Il existe plusieurs outils pour tester le bon fonctionnement de la résolution des noms :

ping

La commande « **ping** » est la plus simple (mais la plus limité). Elle permet de tester la résolution du nom, mais pas la résolution inverse :

```
$ ping NomDuServeur
```

Host

La commande « **host** », permet de tester la résolution du nom et la résolution inverse :

```
$ host NomDuServeur
```

ou :

```
$ host AdresseIPduServeur
```

nslookup

La commande « nslookup » du paquet « dnsutils », permet également de tester la résolution du nom et la résolution inverse :

```
$ nslookup NomDuServeur
```

```
$ nslookup AdresseIPduServeur
```

dig

La commande « **dig** » du paquet « **dnsutils** », permet également de tester la résolution du nom et la résolution inverse. Mais la commande « **dig** » permet surtout d'interroger directement le serveur bind9 et obtenir de nombreuses autres informations :

```
$ dig NomDuServeur.NomDuDomaine
```

Remarque : Le nom du domaine est obligatoire pour obtenir une réponse (ANSWER SECTION)

ou :

```
$ dig -x AdresseIPduServeur
```

Remarque : Le paramètre « -x » est obligatoire pour obtenir une réponse (ANSWER SECTION).

Wireshark

Une capture de trame à l'aide du logiciel Wireshark permettra de détailler les différentes procédures d'une requête de recherche de nom.

Vérification du bon fonctionnement du service :

sudo named-checkconf -z

```
root@Ubuntu-SERVER-64:/home/fbernier# sudo named-checkconf -z
/etc/bind/db.ghost.fr:12: file does not end with newline
zone ghost.fr/IN: loaded serial 2013022801
zone 205.25.172.in-addr.arpa/IN: loaded serial 2013022801
zone localhost/IN: loaded serial 2
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
root@Ubuntu-SERVER-64:/home/fbernier#
```

named -g

```
root@Ubuntu-SERVER-64:/home/fbernier# named -g
03-Mar-2013 20:47:52.781 starting BIND 9.8.1-P1 -g
03-Mar-2013 20:47:52.781 built with '--prefix=/usr' '--mandir=/usr/share/man' '--infodir=/usr/share/info' '--sysconfdir=/etc/b
ind' '--localstatedir=/var' '--enable-threads' '--enable-largefile' '--with-libtool' '--enable-shared' '--enable-static' '--wi
th-openssl=/usr' '--with-gssapi=/usr' '--with-gnu-ld' '--with-geoip=/usr' '--enable-ipv6' 'CFLAGS=-fno-strict-aliasing -DDIG_S
IGCHASE -O2'
03-Mar-2013 20:47:52.781 adjusted limit on open files from 4096 to 1048576
03-Mar-2013 20:47:52.781 found 2 CPUs, using 2 worker threads
03-Mar-2013 20:47:52.782 using up to 4096 sockets
*03-Mar-2013 20:47:52.786 loading configuration from '/etc/bind/named.conf'
03-Mar-2013 20:47:52.787 reading built-in trusted keys from file '/etc/bind/bind.keys'
03-Mar-2013 20:47:52.787 using default UDP/IPv4 port range: [1024, 65535]
03-Mar-2013 20:47:52.787 using default UDP/IPv6 port range: [1024, 65535]
03-Mar-2013 20:47:52.788 listening on IPv4 interface lo, 127.0.0.1#53
03-Mar-2013 20:47:52.789 binding TCP socket: address in use
03-Mar-2013 20:47:52.789 listening on IPv4 interface eth0, 10.0.2.15#53
03-Mar-2013 20:47:52.790 binding TCP socket: address in use
03-Mar-2013 20:47:52.790 listening on IPv4 interface eth1, 172.25.205.250#53
03-Mar-2013 20:47:52.791 binding TCP socket: address in use
03-Mar-2013 20:47:52.791 generating session key for dynamic DNS
03-Mar-2013 20:47:52.791 sizing zone task pool based on 7 zones
03-Mar-2013 20:47:52.795 set up managed keys zone for view _default, file 'managed-keys.bind'
03-Mar-2013 20:47:52.795 Warning: 'empty-zones-enable/disable-empty-zone' not set: disabling RFC 1918 empty zones
03-Mar-2013 20:47:52.795 automatic empty zone: 254.169.IN-ADDR.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 2.0.192.IN-ADDR.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 100.51.198.IN-ADDR.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 113.0.203.IN-ADDR.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 255.255.255.255.IN-ADDR.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
03-Mar-2013 20:47:52.796 automatic empty zone: 1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.IP6.ARPA
03-Mar-2013 20:47:52.797 automatic empty zone: D.F.IP6.ARPA
03-Mar-2013 20:47:52.797 automatic empty zone: 8.E.F.IP6.ARPA
03-Mar-2013 20:47:52.797 automatic empty zone: 9.E.F.IP6.ARPA
03-Mar-2013 20:47:52.797 automatic empty zone: A.E.F.IP6.ARPA
03-Mar-2013 20:47:52.798 automatic empty zone: B.E.F.IP6.ARPA
03-Mar-2013 20:47:52.798 automatic empty zone: 8.B.D.0.1.0.0.2.IP6.ARPA
03-Mar-2013 20:47:52.801 none:0: open: /etc/bind/rndc.key: permission denied
03-Mar-2013 20:47:52.801 couldn't add command channel 127.0.0.1#953: permission denied
03-Mar-2013 20:47:52.801 none:0: open: /etc/bind/rndc.key: permission denied
03-Mar-2013 20:47:52.801 couldn't add command channel ::1#953: permission denied
03-Mar-2013 20:47:52.801 ignoring config file logging statement due to -g option
03-Mar-2013 20:47:52.802 zone 0.in-addr.arpa/IN: loaded serial 1
03-Mar-2013 20:47:52.802 zone 127.in-addr.arpa/IN: loaded serial 1
03-Mar-2013 20:47:52.803 zone 205.25.172.in-addr.arpa/IN: loaded serial 2013022801
03-Mar-2013 20:47:52.804 zone 255.in-addr.arpa/IN: loaded serial 1
03-Mar-2013 20:47:52.805 /etc/bind/db.ghost.fr:12: file does not end with newline
03-Mar-2013 20:47:52.806 zone ghost.fr/IN: loaded serial 2013022801
03-Mar-2013 20:47:52.806 zone localhost/IN: loaded serial 2
03-Mar-2013 20:47:52.807 managed-keys-zone ./IN: loaded serial 3
03-Mar-2013 20:47:52.808 running
```

netstat netstat -ntulp | grep named

```
root@Ubuntu-SERVER-64:/home/fbernier# netstat -ntulp | grep named
tcp        0      0 172.25.205.250:53       0.0.0.0:*               LISTEN      1101/named
tcp        0      0 10.0.2.15:53            0.0.0.0:*               LISTEN      1101/named
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN      1101/named
tcp        0      0 127.0.0.1:953           0.0.0.0:*               LISTEN      1101/named
tcp6       0      0 ::1:953                 :::*                    LISTEN      1101/named
udp        0      0 172.25.205.250:53       0.0.0.0:*                           2668/named
udp        0      0 10.0.2.15:53            0.0.0.0:*                           2668/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           2668/named
udp        0      0 172.25.205.250:53       0.0.0.0:*                           1101/named
udp        0      0 10.0.2.15:53            0.0.0.0:*                           1101/named
udp        0      0 127.0.0.1:53            0.0.0.0:*                           1101/named
```

Tests du fonctionnement de la résolution des noms

Ping , host et nslookup

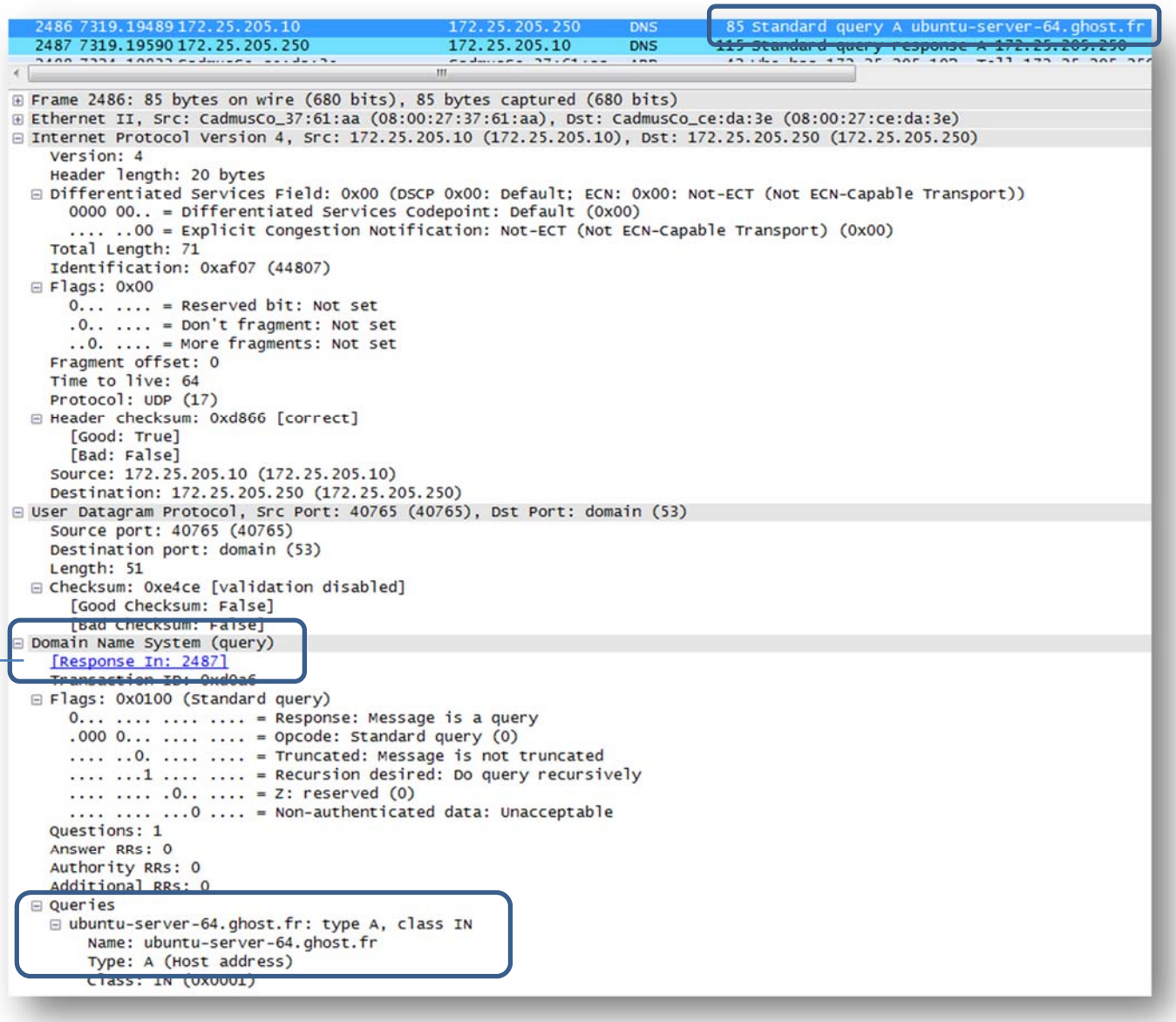
```
root@Ubuntu-SERVER-64:/home/fbernier# ping ubuntu-server-64.ghost.fr -c 4
PING Ubuntu-SERVER-64.ghost.fr (127.0.1.1) 56(84) bytes of data.
64 bytes from Ubuntu-SERVER-64.ghost.fr (127.0.1.1): icmp_req=1 ttl=64 time=0.036 ms
64 bytes from Ubuntu-SERVER-64.ghost.fr (127.0.1.1): icmp_req=2 ttl=64 time=0.060 ms
64 bytes from Ubuntu-SERVER-64.ghost.fr (127.0.1.1): icmp_req=3 ttl=64 time=0.054 ms
64 bytes from Ubuntu-SERVER-64.ghost.fr (127.0.1.1): icmp_req=4 ttl=64 time=0.059 ms

--- Ubuntu-SERVER-64.ghost.fr ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.036/0.052/0.060/0.010 ms
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier# host ubuntu-server-64.ghost.fr
ubuntu-server-64.ghost.fr has address 172.25.205.250
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier#
root@Ubuntu-SERVER-64:/home/fbernier# nslookup ubuntu-server-64.ghost.fr
Server:         172.25.205.250
Address:        172.25.205.250#53

Name:   ubuntu-server-64.ghost.fr
Address: 172.25.205.250

root@Ubuntu-SERVER-64:/home/fbernier#
```

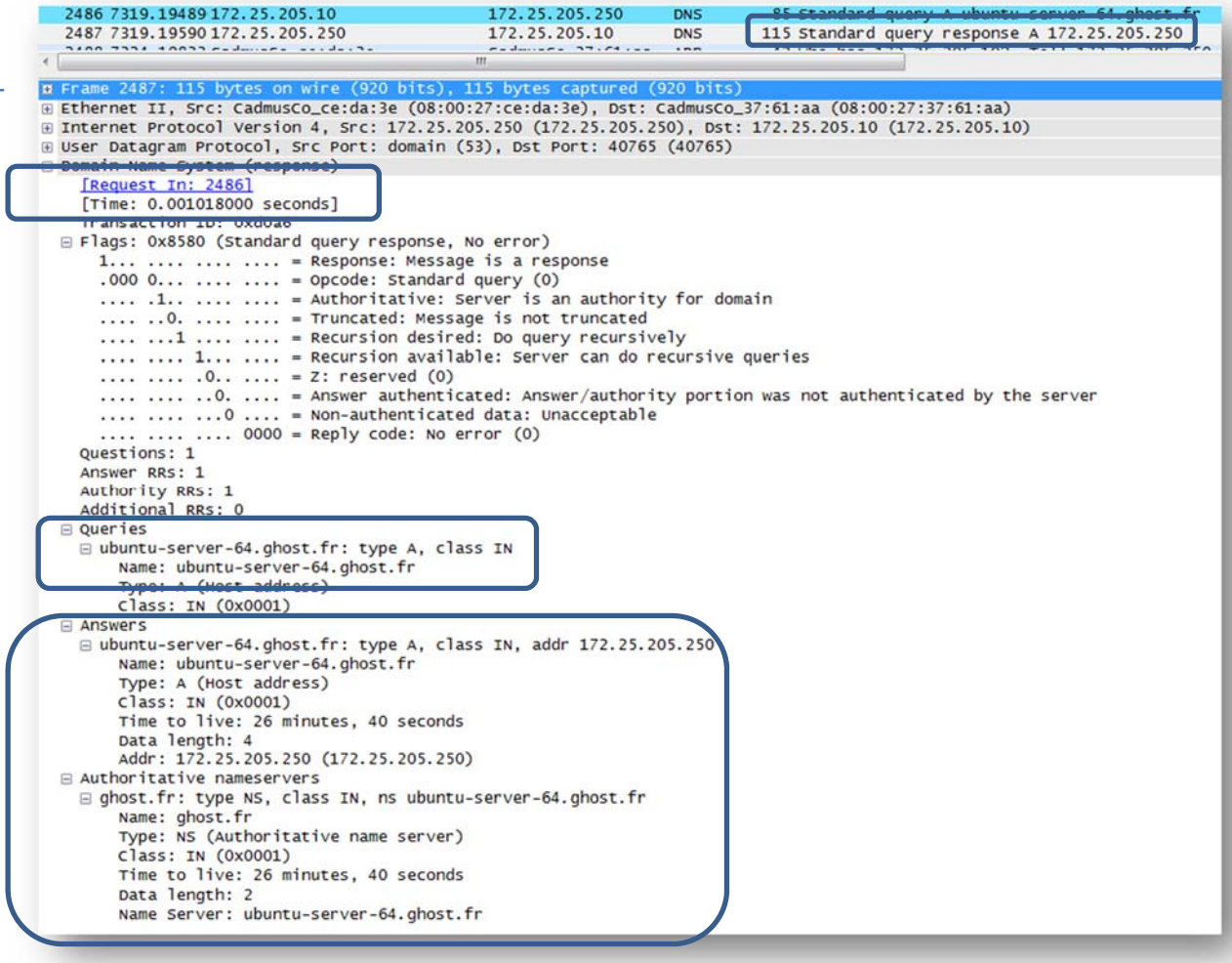
dig

```
root@Ubuntu-SERVER-64:/home/fbernier# dig ubuntu-server-64.ghost.fr

; <<>> DiG 9.8.1-P1 <<>> ubuntu-server-64.ghost.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 54857
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 0

;; QUESTION SECTION:
;ubuntu-server-64.ghost.fr.     IN      A

;; ANSWER SECTION:
ubuntu-server-64.ghost.fr. 1600 IN      A       172.25.205.250

;; AUTHORITY SECTION:
ghost.fr.               1600    IN      NS      ubuntu-server-64.ghost.fr.

;; Query time: 0 msec
;; SERVER: 172.25.205.250#53(172.25.205.250)
;; WHEN: Sun Mar  3 20:55:47 2013
;; MSG SIZE  rcvd: 73

root@Ubuntu-SERVER-64:/home/fbernier#
```

Wireshark

Requête DNS d'un poste client ➔ dig ubuntu-server-64.ghost.fr




2486 7319.19489 172.25.205.10 172.25.205.250 DNS
2487 7319.19590 172.25.205.250 172.25.205.10 DNS 115 Standard query response A 172.25.205.250
Frame 2487: 115 bytes on wire (920 bits), 115 bytes captured (920 bits)
Ethernet II, Src: CadmusCo_ce:da:3e (08:00:27:ce:da:3e), Dst: CadmusCo_37:61:aa (08:00:27:37:61:aa)
Internet Protocol Version 4, Src: 172.25.205.250 (172.25.205.250), Dst: 172.25.205.10 (172.25.205.10)
User Datagram Protocol, Src Port: domain (53), Dst Port: 40765 (40765)
Domain Name System (response)
[Request In: 2486]
[Time: 0.001018000 seconds]
Transaction ID: 0xd0a8
Flags: 0x8580 (Standard query response, No error)
1... .... = Response: Message is a response
.000 0... .... = Opcode: Standard query (0)
.... .1.. .... = Authoritative: Server is an authority for domain
.... ..0. .... = Truncated: Message is not truncated
....1 .... = Recursion desired: Do query recursively
.... 1... .... = Recursion available: Server can do recursive queries
.... .0.. .... = Z: reserved (0)
....0. .... = Answer authenticated: Answer/authority portion was not authenticated by the server
....0 .... = Non-authenticated data: Unacceptable
.... 0000 = Reply code: No error (0)
Questions: 1
Answer RRs: 1
Authority RRs: 1
Additional RRs: 0
Queries
ubuntu-server-64.ghost.fr: type A, class IN
Name: ubuntu-server-64.ghost.fr
Type: A (Host address)
Class: IN (0x0001)
Answers
ubuntu-server-64.ghost.fr: type A, class IN, addr 172.25.205.250
Name: ubuntu-server-64.ghost.fr
Type: A (Host address)
Class: IN (0x0001)
Time to live: 26 minutes, 40 seconds
Data length: 4
Addr: 172.25.205.250 (172.25.205.250)
Authoritative nameservers
ghost.fr: type NS, class IN, ns ubuntu-server-64.ghost.fr
Name: ghost.fr
Type: NS (Authoritative name server)
Class: IN (0x0001)
Time to live: 26 minutes, 40 seconds
Data length: 2
Name Server: ubuntu-server-64.ghost.fr

LEXIQUE

- **$TTL** : (Time To Live) exprime la durée (en secondes) de validité, par défaut, des informations que contiennent les RRs (Les Ressources Records). Une fois ce délai expire, il est nécessaire de revérifier les données. Les différents types :
- **SOA** : permet de définir les informations relatives à la zone. En l'occurrence le nom du serveur DNS primaire "sid.example.com." et l'adresse mail du contact technique (root.example.com. ; le @ est remplacé par un point). Il est compose de plusieurs champs :
- **Serial** : est un entier non signé 32 bits. C'est le numéro de série à incrémenter à chaque modification du fichier. Il permet au serveur secondaire de recharger les informations qu'ils ont. L'usage général vient à le formater de cette manière YYYYMMDDXX, soit pour la première modification du 01/04/2007 → 2007040101, pour la seconde 2007040102.
- **Refresh** : définit la période de rafraîchissement des données.
- **Retry** : si une erreur survient au cours du dernier rafraîchissement, celle -ci sera répéter au bout du délai Retry.
- **Expire**: le serveur sera considéré comme non disponible au bout du délai Expire.
- **Negative cache TTL** : défini la durée de vie d'une réponse NXDOMAIN de notre part.
- **NS** : renseigne le nom des serveurs de noms pour le domaine.
- **MX** : renseigne sur le serveur de messagerie. Plusieurs peuvent être définis. Ainsi, il est possible de leur donner une priorité en leur affectant un numéro. Plus bas est le numéro, plus haute est la priorité.
- **A** : associe un nom d'hôte à une adresse ipv4 (32 bits)
- **AAAA** : associe un nom d'hôte à une adresse ipv6 (128 bits)
- **CNAME** : identifie le nom canonique d'un alias (un nom pointant sur un autre nom)
- **PTR** : c'est simplement la résolution inverse (le contraire du type A).
- **Les classes** :
- **IN** détermine l'association a la classe Internet. D'autres classe sont disponibles (CH et HS). Pour de plus amples informations vous pouvez consulter la http://www.ietf.org/rfc/rfc1035.txt RFC 1035
- **FQDN *fully qualified domain name*** (**FQDN**, ou nom de domaine complètement qualifié) est un nom de domaine qui indique la position absolue d'un nœud dans l'arborescence DNS en indiquant tous les domaines de niveau supérieur jusqu'à la racine.