UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Etude de la géstion d'un système de noms de domaine (DNS) à l'Universite du Burundi : emplementation, exploitation et maintenance

Gahimbare, Aloys; Niyongabo, Hypolite; Sous la direction de : Ir HAKIZIMANA Constaque; Ir VYUMVUHORE Jérôme

2009
UB, Institut Technique Supérieur
https://repository.ub.edu.bi/handle/123456789/1097

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

REPUBLIQUE DU BURUNDI
UNIVERSITE DU BURUNDI
INSTITUT TECHNIQUE SUPERIEUR
DEPARTEMENT DE GENIE ELECTROMECANIQUE



ETUDE DE LA GESTION D'UN SYSTEME DE NOMS DE DOMAINE (DNS) A L'UNIVERSITE DU BURUNDI : IMPLEMENTATION, EXPLOITATION ET MAINTENANCE

Par :

GAHIMBARE Aloys
NIYONGABO Hypolite

Sous la direction de :
Ir HAKIZIMANA Constaque
Ir VYUMVUHORE Jérôme

Mémoire présenté et soutenu publiquement en vue de l'obtention du grade d'ingénieur industriel en génie électromécanique

Bujumbura, Août 2009

DEDICACES

Au Dieu Tout-Puissant,

A vous chers parents,

A vous chers frères et sœurs,

A vous chers oncles et tantes,

A vous chers amis et connaissances,

A tous ceux qui me sont chers et précieux

Je dédie ce mémoire.

Hypolite NIYONGABO

Au Dieu Tout-Puissant,

A vous chers parents,

A vous chers frères et sœurs,

A vous chers oncles et tantes,

A mes neveux et nièces,

A mes cousins et cousines,

A tous ceux qui me sont chers et à qui je suis cher,

Je dédie ce mémoire.

Aloys GAHIMBARE

REMERCIEMENTS

Un travail de recherche est une entreprise laborieuse qui requiert le soutien et le concours d'un certain nombre de personnes sans lesquelles le travail n'arriverait pas à terme. L'occasion nous est donc offerte ici pour exprimer notre gratitude à toute personne qui, de près ou de loin, a contribué pour mener à bon port notre tâche.

Notre profonde reconnaissance s'adresse avant tout au Dieu Tout Puissant. Son amour nous a permis d'affronter toutes les épreuves pour arriver jusqu'à cette date. Que son nom soit toujours loué et glorifié.

Nos remerciements les plus appuyés vont à l'endroit des ingénieurs HAKIZIMANA Constaque et VYUMVUHORE Jérôme respectivement directeur et codirecteur de ce mémoire qui, malgré leur emploi du temps chargé, n'ont pas manqué de nous aider, de nous guider et de nous inculquer l'amour du travail bien fait. La patience, l'abnégation et la disponibilité dont ils ont fait preuve les honorent et font d'eux des éducateurs hors pair. Nous profitons de cette opportunité pour leur exprimer nos sentiments de gratitude, d'estime et de profond respect.

Nous ne manquerons pas non plus de rendre un vibrant hommage à nos familles qui, depuis notre tendre enfance, n'ont pas cessé de nous prodiguer des conseils, de nous encourager et de développer en nous le sens du travail et de la persévérance.

Que les familles de RUBAVU Charles, RUKENKANYA Adolphe, NIKOBIRI Christophe, SOKOROZA Daniel, MATENDEKANO Mathieu, BIGIRIMANA Servat et NIMENYA Cyrile trouvent ici le couronnement de leurs efforts et de leur soutien. Votre assistance et l'affection dont vous nous avez témoigné resteront à jamais gravées dans nos mémoires.

Nous n'oublierons pas de remercier aussi les cadres de l'Université du Burundi et plus particulièrement Dr Rachèle AKIMANA, chef de service informatique, qui nous a donné l'accès au réseau du l'Université et l' Ir Evelyne NDAYISHIMIYE pour son encadrement. Nos vifs remerciements vont aussi à l'endroit de l'Ir Ephrem SEBATIGITA. Ses conseils et ses observations nous ont permis de comprendre les méandres du domaine que nous avons traité.

Enfin, nous rendons sincèrement un vibrant hommage aux éducateurs qui nous ont instruits depuis l'école primaire jusqu'aux études supérieures et plus particulièrement ceux de l'Institut Technique Supérieur à l'Université du Burundi pour la formation tant morale qu'intellectuelle reçue de leur part.

A tous et à chacun, nous disons merci.

GAHIMBARE Aloys & NIYONGABO Hypolite

LISTE DES SIGLES ET ABREVIATIONS

ARPANET	: Advanced Research Projects Agency Network
CPU	: Central Processing Unit
ccTLD	: Country Code Top Level Domain
DNS	: Domain Name System
FQDN	: Fully Qualified Domain Name
gTLD	: Generic Top Level Domain
IP	: Internet Protocol
IPV 4	: Internet Protocol Version 4
IPV 6	: Internet Protocol Version 6
ISP	: Internet Service Provider
NTP	: Network Time Protocol
OS	: Open Source
PTR	: Pointeur
RR	: Resource Record
TLD	: Top Level Domain
TTL	: Time To Live
TCP/IP	: Transfer Control Protocol/Internet Protocol
TDD	: Time Division Duplex

LISTE DES FIGURES

Figure 1: Hiérarchie d'un espace de nom de domaine. 11
Figure 2 : Schéma d'une requête récursive 28
Figure 3 : Schéma d'une requête itérative 31
Figure 4 : Schéma des différentes pannes du DNS 58
Figure 5: Schéma des vulnérabilités résolues et non résolues par TSIG 61
Figure 6 : Schéma de bilan des vulnérabilités résolues par DNSSec 68
Figure 7 : Schéma du réseau wireless de l' Université du Burundi de 2004 à Juillet 2007 73
Figure 8: Antenne VSAT bande C de l'Université du Burundi 74
Figure 9 : Figure de la station de base de l'Université du Burundi 75
Figure 10 : Structure du réseau wireless de l'Université du Burundi après 76

TABLE DES MATIERES

DEDICACES i
REMERCIEMENTS iii
LISTE DES SIGLES ET ABREVIATIONS v
LISTE DES FIGURES vi
TABLE DES MATIERES vii

CHAPITRE 0 : INTRODUCTION 1

0.1. Intérêt du sujet 1
0.2. Délimitation du sujet et méthodologie 2
0.3. Articulation du travail 3

CHAPITRE I : HISTORIQUE ET PRINCIPES DU DNS 4

INTRODUCTION 4
I.1. Historique du système de noms de domaine 5
I.2. Présentation du système de noms de domaine 7
I.3. La hiérarchie du système de noms de domaine 8
I.4. Définition et principes du DNS 9
I.4.1. Définition du DNS 9
I.4.2. Principes du DNS 10
I.4.3. Les serveurs de noms et de zones 12
I.4.4. Les différents types de zone 16
I.4.5. Le transfert de zone 19
I.4.6. L'ajustement des valeurs de Time To Live 20
I.4.7. La résolution des noms de domaine 22
I.4.8. Les résolveurs 24

I.5. Le choix du nom de domaine **25**

I.6. Les différents enregistrements d'un DNS **25**

I.7. Les différentes requêtes effectuées sur un DNS **28**

I.7.1. La requête récursive 28
I.7.2. La requête itérative 31
I.7.3. La requête inverse 32

I.8. Les types de domaines DNS **33**

I.8.1. Le domaine racine 33
I.8.2. Le domaine de premier niveau 33
I.8.3. Le domaine de second niveau 35
I.8.4. Les sous-domaines 35
I.8.5. Les noms d'hôtes 36

I.9. Le redirecteur et les serveurs esclaves **36**

CHAPITRE II : IMPLEMENTATION DU DNS **38**

II.1. La structure des fichiers DNS **38**

II.1.1. Le fichier de zone 39

II.2. L'initialisation des données du DNS **47**

II.3. Le démarrage d'un serveur primaire **47**

II.4. Le démarrage d'un serveur-esclave **48**

CHAPITRE III : EXPLOITATION ET MAINTENANCE DU DNS **50**

III.1. L'exploitation du DNS **50**

III.1.1. L'expansion de domaine : le choix du nombre de serveurs 50
III.1.2. L'ajout des serveurs 51
III.1.3. La gestion des sous-domaines 52

III.2. La maintenance du DNS 57

III.2.1. La sécurisation des transactions avec TSIG 59
III.2.2. Anticipation de pannes 68
III.2.3. Le traitement de pannes 69

CHAPITRE IV : IMPLEMENTATION DU DNS A L'UNIVERSITE DU BURUNDI 72

IV.1. Situation géographique de l'université du Burundi 72

IV.2. Présentation de l'évolution du réseau informatique de l'Université du Burundi 72

IV.3. Implémentation, exploitation et maintenance du serveur de noms de domaine 77

IV.4. Les principaux outils pour tester le bon fonctionnement d'un réseau83

IV.4.1. L'outil ping 83
IV.4.2. L'outil nslookup 84
IV.4.3. L'outil Traceroute 84
IV.4.4. Les outils ifconfig et netstat 85
IV.4.5. Les utilitaires FTP et TELNET 86

CONCLUSION GENERALE 87

REFERENCES BIBLIOGRAPHIQUES 91

CHAPITRE 0 : INTRODUCTION

0.1. Intérêt du sujet

Dans le domaine de l'informatique les réseaux jouent un rôle très important.
C'est à travers ces infrastructures que se font les échanges de fichiers et documents nécessaires au travail quotidien de ceux qui les utilisent.
Ce sont les services que rendent les réseaux qui ont permis un développement spectaculaire de ces derniers.
Ce développement se caractérise par l'augmentation du nombre de machines connectées au même réseau et l'interconnexion de plusieurs réseaux entre eux. L'Internet qui est un réseau mondial est le cas de cette interconnexion des réseaux.
De nos jours, le monde entier suit de près l'évolution des réseaux informatiques car il s'est avéré que c'est un système incontournable pour l'optimisation du rendement au travail. Au Burundi, la mise en place de réseaux dans presque toutes les institutions est quasi inexistante et là où le processus a commencé, il est encore en chantier.
Ce développement des réseaux informatiques requiert la mise en place d'autres services connexes au domaine. L'exemple type est le serveur de noms.

Au sein d'un réseau, les machines communiquent entre elles par des adresses IP. Au fur et à mesure que le réseau s'accroît il devient difficile aux usagers de localiser géographiquement les machines du réseau ; ce qui limite son utilisation optimale, étant donné qu'il est difficile de mémoriser les chiffres.

C'est ainsi qu'un système permettant d'établir la correspondance entre un nom d'un ordinateur ou d'un autre équipement informatique et son adresse IP s'avère nécessaire.
Pour notre travail de fin d'étude, cet aspect de l'informatique nous a intéressé car pas mal de réseaux informatiques au Burundi en général et à l'Université du Burundi en particulier est en plein chantier. De plus, certains administrateurs et usagers ne sont pas bien informés sur l'utilité de ce service ce qui fait qu'ils se rappellent de sa mise en place un peu tardivement. Pourtant un système de nom de domaine est un outil incontournable pour pouvoir administrer efficacement un réseau.

Ces éléments nous ont poussé à choisir pour notre travail de fin d'études universitaires le thème : ***« Etude de la gestion d'un système de noms de domaine à l'Université du Burundi : implémentation, exploitation et maintenance. »***

0.2. Délimitation du sujet et méthodologie

Notre travail de fin d'étude porte sur le réseau de l'Université du Burundi. Ce réseau a retenu notre attention en raison de son extension ces derniers temps ; de plus il est encore en chantier.
Le jour du choix de notre sujet, le service DNS n'était pas déjà implémenté, il ressort de notre contribution.

Pour mener au bon port notre étude, nous avons utilisé des ouvrages traitant les notions d'informatique, des sites internet traitant la notion de DNS.
Pour l'implémentation du système de noms à l'Université du Burundi, nous avons fait recourt au programme Bind9 tournant sous Linux.

0.3. Articulation du travail

Le présent travail s'articule autour de quatre chapitres précédés d'une introduction et terminés par une conclusion. Notre étude débute par la revue des aspects historiques du DNS et ses principes de base au premier chapitre.

Le second chapitre développe les théories relatives à l'implémentation du système de noms de domaine.

Le troisième chapitre traite les aspects de l'exploitation et la maintenance du DNS dans ses détails.

Le quatrième et dernier chapitre de ce travail retrace l'évolution du réseau informatique de l'Université du Burundi et nous mène à l'implémentation - exploitation du système de nom de domaine au sein de cette université.

Des tests de bien fonctionnement du nouveau système ont été menés en faisant recourt aux outils d'administration d'un réseau informatique. Les résultats de ces tests ont été consignés dans les annexes de ce travail.

CHAPITRE I : HISTORIQUE ET PRINCIPES DU DNS

INTRODUCTION

Depuis environ deux décennies, les réseaux se sont étendus d'une manière incroyable. . En effet, à la fin des années 1986, l'Internet, le réseau mondial le plus important, est passé de quelques 6000 ordinateurs à plus de 600000[1] cinq ans plus tard.

De nature, l'informatique s'occupe de la gestion et du traitement de l'information. Mais cette information serait inutile si elle n'était pas partagée avec d'autres personnes. Le réseau est ainsi le véhicule qui permet aux données d'être facilement partagées. Cela sous-entend la communication entre ordinateurs identifiables sur le réseau par leurs adresses IP codées sur 32 bits (IPV4) ou sur 48 bits (IPV6). Mais ces chiffres étant longs avec aussi la croissance exponentielle de l'Internet, il est difficile voire non pratique à l'être humain de les mémoriser. C'est pour pallier à ce problème qu'a été introduite la notion de système de noms de domaines ; en anglais Domain Name System (DNS).

Dans ce chapitre, nous passerons en revue l'histoire du DNS et de ses principes.

[1]M LOTTOR (1981-1991), RFC 1296, Internet Growth, SRI International

I.1. Historique du système de noms de domaine

Les ordinateurs connectés au réseau Internet communiquent en utilisant un protocole appelé TCP/IP. Il définit un système d'adressage permettant d'identifier chaque ordinateur connecté au réseau à travers une adresse numérique de 32 ou 48bits.

Pour faciliter leur manipulation, ces adresses sont représentées dans un format appelé notation décimale pointée.

La notation numérique pour identifier les machines n'est pas commode pour l'utilisateur qui a du mal à retenir une longue suite de chiffres. Il est plus aisé de leur attribuer des noms qui vont servir à les identifier.

Les machines ne communiquant entre elles qu'avec leurs adresses numériques, il est donc nécessaire de disposer d'un service de mise en correspondance nom - adresse numérique (adresse IP) qui permet aussi bien de trouver l'adresse IP d'une machine à partir de son nom (résolution de nom) que de retrouver le nom à partir de l'adresse IP (résolution inverse).

Pendant les premières années d' Internet, ce service de mise en correspondance était assuré à travers un fichier unique centralisé dans lequel étaient mentionnés les noms de toutes les machines connectées au réseau ainsi que leurs adresses IP correspondantes. Ce fichier dont le nom était **hosts.txt** devait être téléchargé par FTP et stocké dans un fichier standard (**/etc/hosts**) consulté pour effectuer la résolution de nom et la résolution inverse.

Ce procédé, qui était relativement efficace dans les années 70 (avec seulement quelques centaines d'ordinateurs connectés au réseau), a vite atteint ses limites avec l'expansion du réseau auquel sont aujourd'hui connectées plus de cent millions de machines.

Ses limites étaient entre autres :

- Le temps de diffusion des informations assez élevé;
- Une forte probabilité de collisions de noms (tous les noms sont enregistrés dans un domaine unique : **arpa**);
- Le caractère sommaire des informations disponibles dans le fichier (simple mise en correspondance entre noms et adresses IP) ;
- La mise à jour des fichiers : il fallait retransmettre le fichier de mise à jour à tous les hôtes, ce qui encombrait fortement la bande passante ;
- L'autonomie des organismes : avec l'évolution de l'Internet, les architectures ont été transformées, ainsi les organismes locaux ont eu la possibilité de créer leurs propres noms et adresses, et ils étaient alors obligés d'attendre que l'organisme centralisateur prenne en compte leurs nouvelles adresses avant que les sites ne puissent être visibles par tous sur Internet. Le souhait était alors que chacun puisse gérer ses adresses avec une certaine autonomie.

Jusqu'alors les autorités du premier grand réseau (ARPANET) lancent une réflexion sur le développement d'un remplaçant à HOSTS.TXT. Le but était de créer un système résolvant les problèmes inhérents à un fichier central de description des hôtes.

Le nouveau système devrait permettre la gestion locale des données tout en les rendant globalement accessibles.

La décentralisation de la gestion éliminerait le goulot d'étranglement de l'hôte unique est réduirait le problème du trafic. La gestion locale permettrait également d'avoir des données à jour plus facilement. L'espace de noms serait hiérarchique ce qui garantirait l'unicité des noms.

En 1983-1984, Paul Mockapetris et John Postel proposent et développent une solution qui utilise des structures de base de données distribuée : le Domain Name System (DNS). Les spécifications des DNS ont été établies en 1987.

I.2. Présentation du système de noms de domaine

Le DNS est basé sur un modèle en arborescence similaire à celui des systèmes de fichiers et de répertoires, avec une gestion décentralisée des données. Chaque site est responsable des données de sa zone. Il permet de fournir des informations supplémentaires telles que celles concernant le temps de validité des informations, les relais de messagerie, les alias de machines, et assure une mise en correspondance dynamique entre les noms et les adresses IP. Du fait de sa structure hiérarchique, le DNS ramène le domaine de collision, qui était dû au téléchargement par FTP du fichier contenant les données à travers le réseau, à des espaces plus réduits.

Le DNS est donc une base de données distribuée qui permet à certaines machines de contrôler certains segments de la base de données, tandis que toute la base de données est accessible avec un mécanisme client-serveur. Un système de réplication assure une fiabilité raisonnable, tandis qu'un système de cache (mémoire des données récentes) permet d'augmenter la performance du système.

I.3. La hiérarchie du système de noms de domaine

Les noms de machines utilisant le système DNS sont appelés **noms d'hôte**. Un nom d'hôte peut contenir jusqu'à 255 caractères alphanumériques (chiffres et lettres) et le caractère trait d'union "-". L'utilisation du caractère "." est interdite car il est réservé afin de séparer un domaine supérieur d'un domaine inférieur.

En effet, on distingue deux types de noms avec le système DNS :

- le **nom d'hôte** qui représente le nom d'une machine (un ordinateur, une imprimante ou bien encore un routeur) ;
- le **nom de domaine pleinement qualifié ou FQDN** (Fully Qualified Domain Name).

Le FQDN est en fait composé de deux parties : le nom d'hôte et le suffixe DNS. Le suffixe DNS défini la relation entre le domaine auquel il appartient, la machine et le domaine racine.

I.4. Définition et principes du DNS

I.4.1. Définition du DNS

Le Domain Name System (DNS), ou système de noms de domaine en français, est un système qui permet 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. Ce système a été inventé par Paul Mockapetris en 1983.

Un nom de domaine quant à lui est un espace de communication permettant à une société, un organisme, une association ou un particulier de communiquer efficacement sur ses activités, ses produits ou services en valorisant son nom ou sa marque.

De ces deux définitions ressortent le rôle capital du DNS à savoir celui de la résolution des adresses IP, plus difficile à retenir pour l'homme, en noms facilement mémorisables.

I.4.2. Principes du DNS

I.4.2.1. L'espace de noms de domaine

Un espace de noms DNS comprend le domaine racine, des domaines de niveau supérieur, des domaines de niveau secondaire et éventuellement des sous-domaines. La combinaison de l'espace de noms DNS et du nom d'hôte constitue le nom de domaine pleinement qualifié (FQDN, Fully Qualified Domain Name). L'espace de noms DNS permet d'organiser les noms affichés des ressources en une structure logique, facile à comprendre.

La structure hiérarchique de l'espace de noms DNS simplifie considérablement l'organisation et la recherche des ressources.

Les noms inscrits dans la base de données DNS constituent une arborescence logique appelée espace de noms de domaine. Le nom de domaine identifie la position d'un domaine par rapport à son domaine parent dans l'arborescence.

Pour utiliser et administrer un service DNS, l'espace de noms de domaine fait référence à l'intégralité de la structure d'un nom de domaine, de la racine au niveau supérieur de l'arborescence jusqu'aux branches de bas niveau.

L'arborescence doit être conforme aux conventions acceptées pour la représentation des noms DNS. La convention principale est simple : pour chaque domaine, un point est utilisé afin de séparer chaque sous-domaine de son domaine parent, de bas en haut dans l'arborescence.

On appelle domaine toute arborescence ou sous-arborescence se trouvant dans l'espace de noms de domaine.

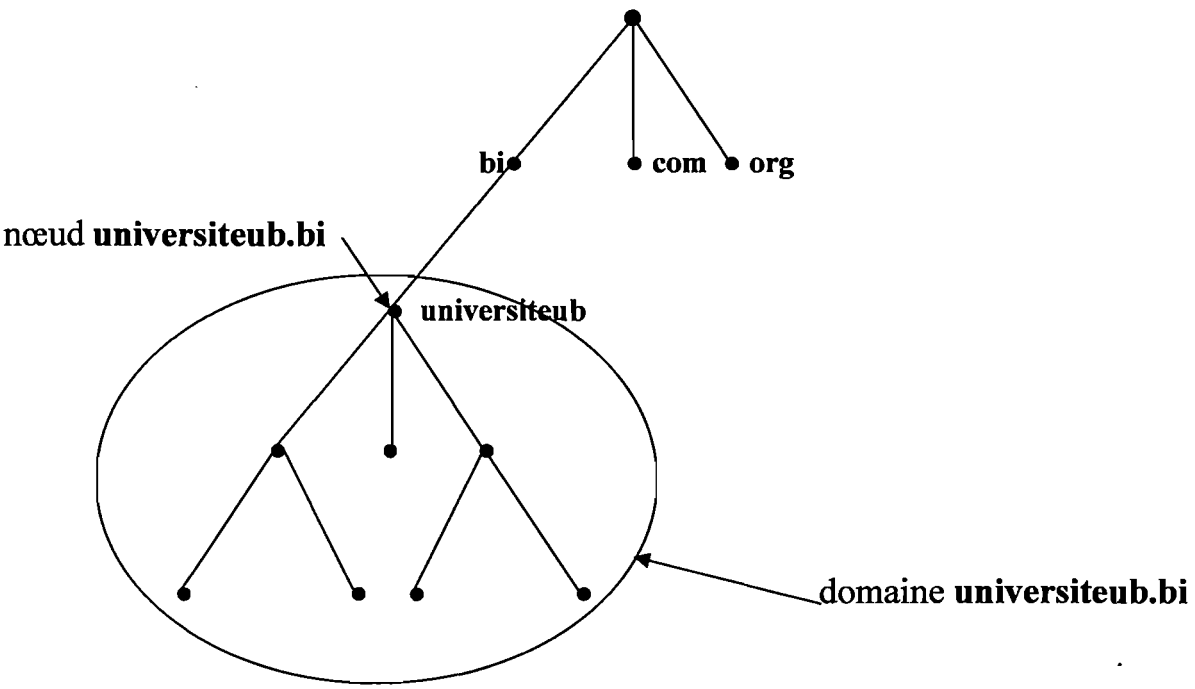


Figure 1: Hiérarchie d'un espace de nom de domaine.

I.4.2.2. La délégation d'autorité

La délégation est le transfert de la responsabilité d'un sous domaine vers un autre organisme. En effet, comme tout dirigeant d'un grand projet peut scinder ce dernier en petites tâches et déléguer la responsabilité de chacune d'elles à différentes personnes, de la même manière, un organisme gérant un domaine peut le diviser en sous domaines dans le but de pouvoir décentraliser son administration. C'est d'ailleurs l'un des principaux buts du système de noms de domaine.

Chacun de ces sous domaines peut être délégué à un autre organisme qui devient par conséquent responsable de la gestion de toutes les informations de ce premier. Cet organisme peut librement modifier les données et même découper son sous domaine en plusieurs sous domaines, puis les déléguer à d'autres responsables. Le domaine parent contient seulement des pointeurs vers les origines du sous domaine et il peut indiquer ces pointeurs à ceux qui les demandent. Tous les organismes ne délèguent pas la totalité de leur domaine, de même que tous les dirigeants ne délèguent pas tout leur travail.

I.4.3. Les serveurs de noms et de zones

Les machines appelées serveurs de nom de domaine permettent d'établir la correspondance entre le nom de domaine et l'adresse IP des machines d'un réseau.

Chaque domaine possède un serveur de noms de domaines, appelé « serveur de noms primaire » (*primary domain name server*), ainsi qu'un serveur de noms secondaire (*secondary domaine name server*), permettant de prendre le relais du serveur de noms primaire en cas d'indisponibilité.

Chaque serveur de nom est déclaré dans un serveur de nom de domaine de niveau immédiatement supérieur, ce qui permet implicitement une délégation d'autorité sur les domaines. Le système de nom est une architecture distribuée, où chaque entité est responsable de la gestion de son nom de domaine. Il n'existe donc pas d'organisme ayant à charge la gestion de l'ensemble des noms de domaines.

Les serveurs correspondant aux domaines de plus haut niveau (TLD) sont appelés « **serveurs de noms racine** ». Il en existe treize, répartis sur la planète, possédant les noms « a.root-servers.net » à « m.root-servers.net ».

Un serveur de noms définit une zone, c'est-à-dire un ensemble de domaines sur lequel le serveur a autorité. Le système de noms de domaine est transparent pour l'utilisateur, néanmoins il ne faut pas oublier les points suivants :

- Chaque ordinateur doit être configuré avec l'adresse d'une machine capable de transformer n'importe quel nom en une adresse IP. Cette machine est appelée Domain Name Server. Lorsque vous vous connectez à Internet, le fournisseur d'accès va automatiquement modifier vos paramètres réseau pour mettre à votre disposition ses serveurs de noms.

- L'adresse IP d'un second *Domain Name Server* (secondary Domain Name Server) doit également être définie : le serveur de noms secondaire peut relayer le serveur de noms primaire en cas de dysfonctionnement.

I.4.3.1. Le serveur de noms primaire

Le serveur primaire est un serveur d'autorité sur sa zone : il tient à jour un fichier appelé "fichier de zone", qui établit les correspondances entre les noms et les adresses IP des hôtes de sa zone. Chaque domaine possède un et un seul serveur primaire.

I.4.3.2. Le serveur de noms secondaire

Un serveur de nom secondaire obtient les données de zone via le réseau à partir d'un autre serveur de nom qui détient l'autorité pour la zone considérée. L'obtention des informations de zone via le réseau est appelée transfert de zone.

Il est capable de répondre aux requêtes de noms (partage de charge), et de secourir le serveur primaire en cas de panne.

Le nombre de serveurs secondaires par zone n'est pas limité. Ainsi il y a une redondance de l'information. Le minimum imposé est un serveur secondaire et le pré requis mais pas obligatoire est de le situer sur un segment différent du serveur primaire.

Un serveur qui effectue un transfert de zone vers un autre serveur est appelé serveur maître. Un serveur maître peut être un serveur primaire ou un serveur secondaire. Un serveur secondaire peut disposer d'une liste de serveurs maîtres (jusqu'à dix serveurs maîtres). Le serveur secondaire contacte successivement les serveurs de cette liste, jusqu'à ce qu'il ait pu réaliser son transfert de zone.

I.4.3.3. Le serveur de noms cache

Un serveur de noms cache est un serveur de noms qui ne fait autorité pour aucune zone. Il émet simplement des requêtes et se souvient du résultat pour une utilisation ultérieure. Pour mettre en place un tel serveur, on configure le serveur de noms comme à l'accoutumé en prenant bien soin de n'inclure aucune zone.

Le serveur cache ne constitue sa base d'information qu'à partir des réponses des serveurs de noms. Il inscrit les correspondances nom / adresse IP dans un cache avec une durée de validité limitée (TTL) ; il n'a aucune autorité sur le domaine : il n'est pas responsable de la mise à jour des informations contenues dans son cache, mais il est capable de répondre aux requêtes des clients DNS.

I.4.3.4. Le serveur DNS racine

Un serveur DNS racine est un serveur DNS qui répond aux requêtes qui concernent le domaine racine et qui les redirige vers le serveur DNS de premier niveau (top-level-domain - TLD) concerné. Bien que n'importe quel opérateur puisse mettre en place ses propres serveurs DNS racine, le terme de "serveur DNS racine" est généralement utilisé pour désigner les treize serveurs DNS racine qui implémentent la racine du "Domain Name System " (système de noms de domaines) officiel d' Internet.

Les serveurs de noms de la racine savent où se trouvent les serveurs de noms de chaque domaine de niveau supérieur. En fait, la plupart des serveurs de la racine font autorité sur les domaines génériques de niveau supérieur.

Pour répondre à une requête concernant un nom quelconque, les serveurs de la racine peuvent au minimum renvoyer les noms et adresses des serveurs faisant autorité pour le domaine de niveau supérieur concernant le nom recherché.

Puis les serveurs de niveau supérieur peuvent renvoyer la liste des serveurs (noms et adresses) faisant autorité pour le domaine de second niveau concernant le nom recherché. Chaque serveur de noms interrogé renvoi soit une indication permettant de poursuivre la recherche, soit la réponse à la requête elle-même.

Les serveurs de noms de la racine ont une importance fondamentale pour la résolution des noms.

Ainsi, le DNS fournit des mécanismes pour limiter leur surcharge, tels que la mémoire cache. Cependant, en l'absence d'indication supplémentaire, la résolution devra commencer au niveau des serveurs de noms de la racine.

Ces serveurs sont cruciaux pour le fonctionnement du DNS, si aucun d'eux n'est accessible sur l'Internet durant une longue période, aucune requête n'aboutira plus. L'Internet dispose de treize serveurs de la racine répartis en différents endroits du réseau.

Étant donnée leur position centrale et malgré leur nombre, les serveurs de la racine sont en permanence occupés et le trafic vers chacun d'eux est très dense. Des mesures récentes ont montré que certains serveurs recevaient plusieurs milliers de requêtes par seconde. Malgré cette charge importante sur les serveurs de la racine, la résolution de noms dans l'Internet fonctionne bien.

I.4.4. Les différents types de zone

Une zone de noms ou zone DNS est un ensemble d'enregistrements de ressources appartenant à la même portion de l'espace de noms DNS. Par exemple une zone DNS peut contenir l'ensemble des enregistrements de ressource de type A (c'est-à-dire des mappages noms d'hôte / adresses IP) du domaine. Il existe trois types de zones DNS : les zones principales, les zones secondaires et les zones de stub.

Les enregistrements d'une zone DNS donnée sont stockés localement par le serveur DNS sous la forme d'un fichier. Cependant si le serveur DNS joue aussi le rôle de contrôleur de domaine, il est possible de stocker les zones principales et les zones de stub dans le service d'annuaire (Active Directory).

On parlera alors de zones intégrées à Active Directory. Cette seconde solution apporte des avantages en termes de performance et de sécurité.

I.4.4.1. Zone principale

Une zone principale est l'exemplaire maître de la zone DNS. Les enregistrements de ressource y sont crées et gérés c'est-à-dire ajoutés, modifiés ou supprimés. Dans une implémentation de DNS classique c'est-à-dire non intégré à Active Directory, un seul serveur de nom agit en tant que serveur principal pour une zone donnée. C'est lui qui détient la copie maîtresse de la zone. Les serveurs qui reçoivent une copie de la zone en lecture seule sont les serveurs secondaires. Le serveur de noms peut héberger plusieurs zones et être à la fois un serveur principal pour une zone et un serveur secondaire pour une autre zone. Dans tous les cas, les modifications d'une zone donnée seront effectuées sur le serveur principal de la zone.

I.4.4.2. Zone secondaire

Une zone secondaire est une copie, en lecture seule, de la zone DNS. Les enregistrements contenus dans la zone secondaire ne peuvent être modifiés.

Une zone secondaire permet d'apporter une redondance en cas de défaillance matérielle et de soulager la charge de travail sur le serveur principal. Les administrateurs peuvent modifier uniquement les enregistrements de la zone DNS principale.

Il est possible de configurer plusieurs serveurs secondaires à des emplacements distants, de telle sorte que les enregistrements de la zone puissent être résolus sans que la requête ne franchisse des liaisons inter sites.

I.4.4.3. Zones de stub

Les zones de stub sont des copies d'une zone qui contiennent uniquement les enregistrements de ressources nécessaires à l'identification du serveur DNS faisant autorité pour la zone en question. Une zone de stub contient un sous ensemble des donnés de la zone qui se compose d'enregistrement SOA, NS et A. Dans ce cas, un enregistrement A est aussi appelé enregistrement glu car il crée une liaison avec la zone déléguée en indiquant l'adresse d'un serveur de noms cité dans un enregistrement NS.

Un serveur hébergeant une zone de stub ne fait pas autorité pour cette zone. Une zone de stub est en quelque sorte un « pointeur » vers le serveur DNS qui fait autorité pour la zone DNS concernée.

I.4.5. Le transfert de zone

Pour chaque zone DNS, le serveur servant de référence est le DNS maître ou DNS primaire. Les DNS esclaves ou secondaires servant cette zone vont récupérer les informations du DNS maître. Cette récupération d'information est appelée transfert de zone.

Seuls les DNS secondaires ont besoin d'être autorisés à effectuer cette opération, mais assez souvent aucune restriction n'est présente. Ceci permettant à n'importe qui de se connecter via nslookup ls -d pour l'affichage du contenu d'une zone.

Lorsque des changements apparaissent sur une zone, il faut que tous les serveurs qui gèrent cette zone en soient informés. Les changements sont effectués sur le serveur principal, le plus souvent en éditant un fichier.

Après avoir édité le fichier, l'administrateur signale au serveur qu'une mise à jour a été effectuée, le plus souvent au moyen d'un signal (SIGINT).

Les serveurs secondaires interrogent régulièrement le serveur principal pour savoir si les données ont changé depuis la dernière mise à jour. Ils utilisent un numéro constitué de la date au format américain: année, mois, jour; version qui est toujours incrémenté. Donc pour la mise à jour ils comparent le champ SERIAL du RR SOA de la zone donnée par le serveur principal contenant le numéro à celui qu'ils connaissent. Si ce numéro a augmenté, ils chargent les nouvelles données.

I.4.6. L'ajustement des valeurs de Time To Live

Le TTL d'un enregistrement de ressources est la durée de maintien de l'enregistrement dans la mémoire cache d'un serveur. Le TTL est exprimée en secondes. Si le TTL d'un enregistrement est de 3600 secondes et qu'un serveur hors du réseau a mémorisé cet enregistrement dans sa mémoire cache, ce serveur devra effacer l'enregistrement au bout d'une heure. S'il a à nouveau besoin de la même information, il devra effectuer une nouvelle recherche.

Le choix d'une valeur de TTL permet d'imposer une politique sur la durée de vie des données aux autres serveurs. Cette politique a un impact sur la charge des serveurs du réseau. Les valeurs de TTL peuvent être modifiées au cour du temps et les administrateurs les ajusteront périodiquement.
Lors d'opérations normales, les serveurs extérieurs au domaine conservent l'adresse de l'hôte dans leur mémoire cache en accord avec le TTL indiqué, soit par la directive $TTL, soit dans l'enregistrement SOA. Il est commode de fixer des valeurs de TTL basses afin que les serveurs distants ne conservent que peu de temps un enregistrement d'adresse. En diminuant le TTL, on force les serveurs externes à mettre à jour leurs données plus souvent.

De cette manière, toutes les modifications effectuées seront propagées plus rapidement. Malheureusement on ne peut pas fixer le TTL à zéro, ce qui indiquerait de ne placer aucun enregistrement en mémoire cache. Par contre, les petites valeurs de TTL ne posent aucun problème.
La méthode la plus simple consiste à modifier la structure de contrôle $TTL.

Si un enregistrement de ressource autre que le SOA ne comporte pas de TTL explicite, le serveur utilise le TTL par défaut pour cet enregistrement spécifique. En réduisant la valeur du TTL par défaut, la nouvelle valeur de TTL s'applique à tous les enregistrements d'adresse et pas seulement à celui concernant l'hôte qui va être déplacé. La conséquence de cette approche est que le serveur va devoir répondre à plus de requêtes que d'ordinaire. Par conséquent, il vaut mieux ne diminuer le TTL que sur l'enregistrement modifié.

Pour ajouter une valeur de TTL explicite dans un enregistrement de ressources, il faut la placer devant IN du champ classe. La valeur du TTL est en secondes en standard, mais on peut aussi préciser des unités (m pour minutes, h pour heures, d pour jours et w pour semaines) sous la même forme que dans la structure de contrôle $TTL.
Un serveur-esclave renvoie les mêmes valeurs de TTL qu'un serveur-maître. L'esclave ne tient pas compte du temps écoulé depuis le dernier chargement de zone pour décrémenter le TTL. Par conséquent, si le TTL d'un enregistrement est réduit à une valeur inférieure au minimum, les serveurs primaires et esclaves fourniront cette valeur réduite. Si un esclave a atteint la limite de validité de la zone, il met la totalité de la zone hors service; il n'invalide jamais isolément un enregistrement de ressource.

La valeur de TTL doit être diminuée bien avant le changement de l'adresse: il ne faut pas réduire simultanément le TTL et modifier l'information, car il se pourrait que l'enregistrement d'adresse vienne d'être stocké dans la mémoire cache d'un serveur distant et il serait valide tant que le TTL d'origine ne se serait pas écoulé. Il faut aussi tenir compte de la fréquence de synchronisation des serveurs esclaves.

Si le TTL minimal est de 12 heures et que l'intervalle de rafraîchissement est de 3heures, il faut réduire le TTL au moins 15heures avant la modification d'adresse, de manière à ce que les enregistrement associés à l'ancien long TTL soient arrivés à expiration.

I.4.7. La résolution des noms de domaine

Le mécanisme consistant à trouver l'adresse IP correspondant au nom d'un hôte est appelé résolution de nom de domaine. L'application permettant de réaliser cette opération (généralement intégrée au système d'exploitation) est appelée résolveur en anglais resolver.

Lorsqu'une application souhaite se connecter à un hôte connu par son nom de domaine, celle-ci va interroger un serveur de noms défini dans sa configuration réseau. Chaque machine connectée au réseau possède en effet dans sa configuration les adresses IP de deux serveurs de noms de son fournisseur d'accès.

Une requête est ainsi envoyée au premier serveur de noms appelé serveur de nom primaire. Si celui-ci possède l'enregistrement dans son cache, il l'envoie à l'application, dans le cas contraire il interroge un serveur racine.

Le serveur de nom racine renvoie une liste de serveurs de noms faisant autorité sur le domaine. Le serveur de noms primaire faisant autorité sur le domaine va alors être interrogé et retourner l'enregistrement correspondant à l'hôte sur le domaine.

I.4.7.1. La résolution directe

La traduction de noms en adresse IP est la résolution directe : **Nom -> Adresse IP.** Dans un réseau IP, pour qu'une machine A puisse communiquer avec une autre machine B grâce au protocole IP, A doit connaître l'adresse IP de B.

Toutefois cette communication ne s'établit qu'à la demande des utilisateurs qui souvent ne savent pas comment les adresses IP sont utilisées ou attribuées aux machines. De plus la mémorisation de ces adresses n'est pas facile pour les humains. D'où l'utilisation des noms ayant un sens significatif .Dans ce cas les machines doivent traduire ces noms en adresses correspondantes utilisables entre elles. C'est le mécanisme de la résolution directe.

I.4.7.2. La résolution inverse

La traduction d'une adresse IP en nom est la résolution inverse:

Adresse IP → nom.

La machine B reçoit un datagramme IP en provenance de A. Ce datagramme contient l'adresse IP de A. La machine B doit donc être capable de trouver le nom FQDN de A à partir de son adresse IP. C'est ce qu'on appelle la résolution de noms inverse.

I.4.8. Les résolveurs

Les "résolveurs" sont des programmes qui interfacent les applications utilisateurs aux serveurs de noms de domaines. En effet, ce n'est pas l'utilisateur qui effectue les requêtes directement. Dans le cas le plus simple, un résolveur reçoit une requête provenant d'une application sous la forme d'un appel d'une fonction de bibliothèque, d'un appel système etc., et renvoie une information sous une forme compatible avec la représentation locale de données du système.

Le résolveur est situé sur la même machine que l'application recourant à ses services, mais devra par contre consulter des serveurs de noms de domaines sur d'autres hôtes.

Comme un résolveur peut avoir besoin de contacter plusieurs serveurs de noms, ou obtenir les informations directement à partir de son cache local, le temps de réponse d'un résolveur peut varier selon de grandes proportions, depuis quelques millisecondes à plusieurs secondes

L'une des raisons les plus importantes qui justifient l'existence des résolveurs est d'éliminer le temps d'acheminement de l'information depuis le réseau, et de décharger simultanément les serveurs de noms, en répondant à partir des données cachées en local. Il en résulte qu'un cache partagé entre plusieurs processus, utilisateurs, machines, etc., sera incomparablement plus efficace qu'une cache non partagé.

I.5. Le choix du nom de domaine

Étant donné que le nom de domaine doit être facile à diffuser, il est indispensable de le choisir le plus simple possible. Il est évident qu'un nom de domaine n'est pas qu'une liste complexe de caractères, il doit être choisi à bon escient en évitant les écueils suivants:

- choisir des noms compliqués ;
- choisir des noms trop longs ;
- mettre des caractères spéciaux.

De plus, le nom de domaine doit:

- être prononçable;
- avoir dans la mesure du possible une signification;
- être disponible.

I.6. Les différents enregistrements d'un DNS

Un DNS est une base de données répartie contenant des enregistrements, appelés **RR** (*Resource Records*), concernant les noms de domaines. Seules sont concernées par la lecture des informations ci-dessous les personnes responsables de l'administration d'un domaine, le fonctionnement des serveurs de noms étant totalement transparent pour les utilisateurs.

U. B.

En raison du système de cache permettant au système DNS d'être réparti, les enregistrements de chaque domaine possèdent une durée de vie, appelée **TTL** (*Time To Live*, traduit *espérance de vie*), permettant aux serveurs intermédiaires de connaître la date de péremption des informations et ainsi savoir s'il est nécessaire ou non de la revérifier.

D'une manière générale, un enregistrement DNS comporte les informations suivantes :

- **Le nom de domaine** : le nom de domaine doit être un nom FQDN, c'est-à-dire être terminé par un point. Si le point est omis, le nom de domaine est relatif, c'est-à-dire que le nom de domaine principal suffixera le domaine saisi ;

- **Le type** : une valeur sur 16 bits spécifiant le type de ressource décrit par l'enregistrement. Le type de ressource peut être un des suivants :

- **A** : il s'agit du type de base établissant la correspondance entre un nom canonique et une adresse IP. Par ailleurs il peut exister plusieurs enregistrements A, correspondant aux différents serveurs du réseau.

- **CNAME** (*Canonical Name*) : il permet de faire correspondre un alias au nom canonique. Il est particulièrement utile pour fournir des noms alternatifs correspondant aux différents services d'une même machine.

- **HINFO** : il s'agit d'un champ uniquement descriptif permettant de décrire notamment le matériel (CPU) et le système d'exploitation (OS) d'un hôte.

Il est généralement conseillé de ne pas le renseigner afin de ne pas fournir d'éléments d'informations pouvant se révéler utiles pour des pirates informatiques.

- **MX** (*Mail eXchange*) : correspond au serveur de gestion du courrier.

Lorsqu'un utilisateur envoie un courrier électronique à une adresse (utilisateur@domaine), le serveur de courrier sortant interroge le serveur de nom ayant autorité sur le domaine afin d'obtenir l'enregistrement MX. Il peut exister plusieurs MX par domaine, afin de fournir une redondance en cas de panne du serveur de messagerie principal. Ainsi l'enregistrement MX permet de définir une priorité avec une valeur pouvant aller de 0 à 65 535

- **NS** : correspond au serveur de noms ayant autorité sur le domaine.

- **PTR** : un pointeur vers une autre partie de l'espace de noms de domaines.

- **SOA** (*Start Of Authority*) : le champ SOA permet de décrire le serveur de nom ayant autorité sur la zone, ainsi que l'adresse électronique du contact technique (dont le caractère « @ » est remplacé par un point).

- **Classe** : la classe peut être soit **IN** correspondant aux protocoles d'Internet, soit **CH** Pour le système chaotique.

- **RDATA** : il s'agit des données correspondant à l'enregistrement.

I.7. Les différentes requêtes effectuées sur un DNS

I.7.1. La requête récursive

Lorsqu' un serveur DNS reçoit une requête récursive, il doit donner la réponse la plus complète possible. C'est pourquoi le serveur DNS est souvent amené à joindre d'autres serveurs de noms dans le but de trouver la réponse exacte.

Le mode récursif est plus simple du point de vue du client. Dans ce mode, le premier serveur prend le rôle de résolveur.

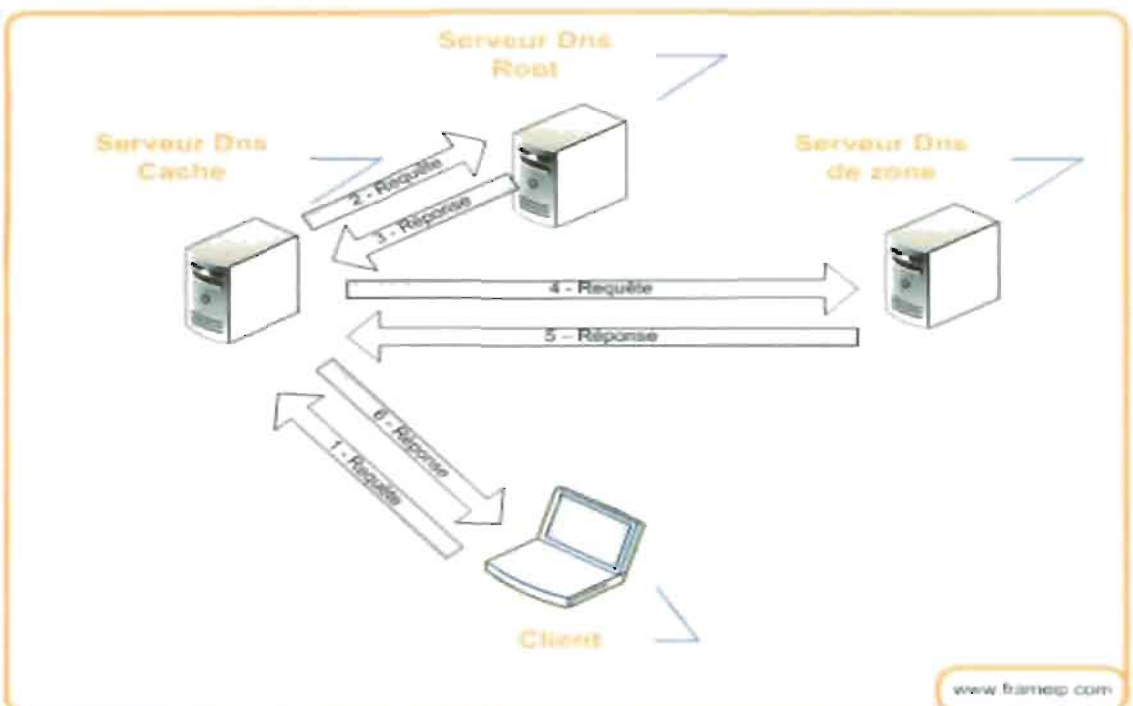


Figure 2 : Schéma d'une requête récursive

L'utilisation du mode récursif est limitée aux cas qui résultent d'un accord négocié entre le client et le serveur. Cet accord est négocié par l'utilisation de deux bits particuliers des messages de requête et de réponse :

Le bit RA (Récursion Admissible), est marqué ou non par le serveur dans toutes les réponses. Ce bit est marqué si le serveur accepte à priori de fournir le service récursif au client, que ce dernier l'ait demandé ou non. Autrement dit, le bit RA signale la disponibilité du service plutôt que son utilisation.

Les requêtes disposent d'un bit RD (pour "récursion désirée"). Ce bit indique que le requérant désire utiliser le service récursif pour cette requête. Les clients peuvent demander le service récursif à n'importe quel serveur de noms, bien que ce service ne puisse leur être fourni que par les serveurs qui auront déjà marqué leur bit RA, ou des serveurs qui auront donné leur accord pour ce service par une négociation propriétaire ou tout autre moyen hors du champ du protocole DNS.

Le mode récursif est mis en oeuvre lorsqu'une requête arrive avec un bit RD marqué sur un serveur annonçant disposer de ce service, le client peut vérifier si le mode récursif a été utilisé en constatant que les deux bits Ra et Rd ont été marqués dans la réponse.

Il est à noter que le serveur de noms ne doit pas utiliser le service récursif s'il n'a pas été explicitement demandé par un bit RD, car cela interfère avec la maintenance des serveurs de noms et de leurs bases de données.

Lorsque le service récursif est demandé et est disponible, la réponse récursive à une requête doit être l'une des suivantes :

- La réponse à la requête, éventuellement préfacée par un ou plusieurs RR CNAME qui indique les alias trouvés pendant la recherche de la réponse ;
- Une erreur de nom indiquant que le nom demandé n'existe pas. Celle-ci peut inclure des RR CNAME qui indiquent que la requête originale pointait l'alias d'un nom qui n'existe pas ;
- Une indication d'erreur temporaire.

Si le service récursif n'est pas requis, ou n'est pas disponible, la réponse non récursive devra être l'une des suivantes :

- Une réponse d'erreur "autorisée" indiquant que le nom n'existe pas ;
- Une indication temporaire d'erreur ;
- Une combinaison :

- des RR qui répondent à la question, avec indication si les données sont extraites d'une zone ou d'un cache ;

- d'une référence à un serveur de noms qui gère une zone plus "proche" du nom demandé que le serveur qui a été contacté ;

- les RR que le serveur de nom pense être utile au requérant pour continuer sa recherche.

I.7.2. La requête itérative

Lorsqu'un serveur reçoit une requête itérative, il renvoie la meilleure réponse qu'il peut donner sans contacter d'autres serveurs DNS (c'est-à-dire en consultant uniquement sa propre base de données).

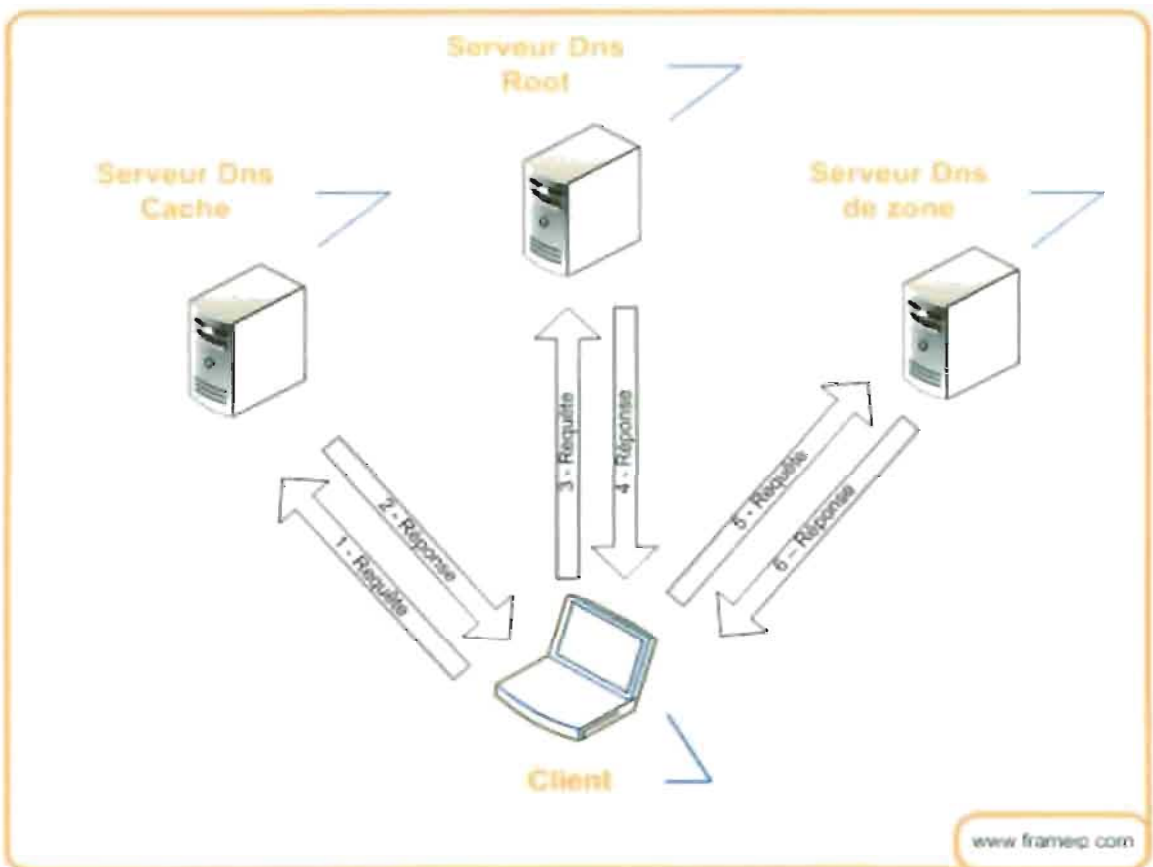


Figure 3 : Schéma d'une requête itérative

Les avantages d'une résolution itérative se présentent :

- dans le cas d'une implémentation simplifiée d'un résolveur qui ne sait exploiter d'autres réponses qu'une réponse directe à la question ;
- dans le cas d'une requête qui doit passer à travers d'autres protocoles ou autres "frontières" et doit pouvoir être envoyée à un serveur jouant le rôle d'intermédiaire ;

- dans le cas d'un réseau dans lequel intervient une politique de cache commun plutôt qu'un cache individuel par client.

Le service non-récursif est approprié si le résolveur est capable de façon autonome de poursuivre sa recherche et est capable d'exploiter l'information supplémentaire qui lui est envoyée pour l'aider à résoudre son problème.

I.7.3. La requête inverse

Dans le cas d'une requête inverse, le résolveur envoie une demande à un serveur de noms afin que celui-ci renvoie le nom d'hôte associé à une adresse Ip connue. C'est utile surtout pour des questions de sécurité, pour savoir avec qui on échange. La mise en place de la résolution inverse est un peu plus compliquée, car l'adressage par nom est basé sur la notion de domaine qui souvent n'a rien à voir avec la structure des adresses IP.
Par conséquent, seule une recherche approfondie portant sur tous les domaines peut garantir l'obtention d'une réponse exacte

I.8. Les types de domaines DNS

Pour comprendre DNS, il faut se souvenir de ce qu'est un nom de domaine entièrement qualifié (FQDN, Fully Qualified Domain Name).Au fait, le nom d'hôte et le nom de domaine combiné forment le FQDN.

Le nom de domaine contient normalement des points (univesiteub.bi, par exemple) qui séparent les niveaux du FQDN. Ces niveaux sont organisés, du plus élevés au plus bas, de la manière suivante :

I.8.1. Le domaine racine

Le domaine racine se situe au sommet de l'arborescence DNS et n'apparaît normalement pas dans le FQDN. Si la racine doit être indiquée, elle l'est par le point final du FQDN. Si elle est seule, elle est indiquée par un point (.).

I.8.2. Le domaine de premier niveau

Il s'agit de la portion finale à droite d'un nom de domaine. En général, un domaine de premier niveau est représenté par un nom de deux ou trois caractères qui identifie le statut organisationnel ou géographique du nom de domaine. Il y' a de nombreux domaines de premier niveau, tels que .com, .edu, .org, .net, .gov, .mil, .arpa, .fr, etc. Les domaines de premier niveau doivent être connus du domaine racine pour que les demandes de noms soient envoyées au serveur de second niveau approprié.

Il existe deux catégories de **TLD** (*Top Level Domain*, soit *domaines de plus haut niveau*) :

➢ Les domaines dits « génériques », appelés **gTLD** (*generic TLD*).

Les gTLD sont des noms de domaines génériques de niveau supérieur proposant une classification selon le secteur d'activité. Ainsi chaque gTLD possède ses propres règles d'accès :

- gTLD historiques :
- **.arpa** correspond aux machines issues du réseau originel ;
- **.com** correspondait initialement aux entreprises à vocation commerciale. Désormais ce TLD est devenu le « TLD par défaut » et l'acquisition de domaines possédant cette extension est possible, y compris par des particuliers.
- **.edu** correspond aux organismes éducatifs ;
- **.gov** correspond aux organismes gouvernementaux ;
- **.int** correspond aux organisations internationales ;
- **.mil** correspond aux organismes militaires ;
- **.net** correspondait initialement aux organismes ayant trait aux réseaux.

Ce TLD est devenu depuis quelques années un TLD courant. L'acquisition de domaines possédant cette extension est possible, y compris par des particuliers ;

- **.org** correspond habituellement aux entreprises à but non lucratif ;
- **.aero** correspond à l'industrie aéronautique ;
- **.biz** (*business*) correspondant aux entreprises commerciales ;
- **.museum** correspond aux musées ;
- **.name** correspond aux noms de personnes ou aux noms de personnages

imaginaires ;

- **.info** correspond aux organisations ayant trait à l'information ;
- **.coop** correspondant aux coopératives ;
- **.pro** correspondant aux professions libérales.
- **gTLD** spéciaux :
- **.arpa** correspond aux infrastructures de gestion du réseau. Le gTLD arpa sert ainsi à la résolution inverse des machines du réseau, permettant de trouver le nom correspondant à une adresse IP.

➢ Les domaines dits «nationaux », appelés **ccTLD** (country code TLD). Les ccTLD correspondent aux différents pays et leurs noms correspondent aux abréviations des noms de pays définies par la norme ISO 3166.

I.8.3. Le domaine de second niveau

Après avoir spécifié le type de système, il faut donner le nom d'une entreprise ou d'une organisation .Les domaines de second niveau eux aussi doivent être connus du domaine de premier niveau.

I.8.4. Les sous-domaines

Outre le nom de second niveau, une entreprise peut choisir de subdiviser encore son nom de domaine en ajoutant des départements ou des services représentés chacun par une portion distincte dans le nom de domaine. C'est ce que permet la création de sous-domaines.

I.8.5. Les noms d'hôtes

Le nom de l'ordinateur, qui est résolu en adresse IP, constitue la fin du FQDN. Aux origines de TCP/IP,étant donné que les réseaux étaient très peu étendus ou autrement dit que le nombre d'ordinateurs connectés à même réseau était faible, les administrateurs réseau créaient des fichiers appelés tables de conversion manuelle.

Ces tables de conversion manuelle étaient des fichiers séquentiels, généralement nommés hosts ou hosts.txt, associant sur chaque ligne l'adresse IP de la machine et le nom littéral associé, appelé nom d'hôte.

I.9. Le redirecteur et les serveurs esclaves

Lorsqu'un client contacte un serveur DNS pour résoudre un nom, le serveur DNS consulte d'abord ses fichiers locaux. S'il n'a pas autorité sur la zone couverte par la requête, il doit s'adresser à un autre serveur de noms pour résoudre la requête. Quand on explore le web, la résolution d'un nom de domaine peut impliquer d'envoyer une requête au serveur de noms de votre fournisseur d'accès ou de contacter un autre serveur de noms de l'Internet.

On peut ne pas souhaiter que tous les serveurs DNS propagent ces requêtes. DNS Microsoft permet de désigner certains serveurs DNS comme redirecteurs.

En général, seuls les redirecteurs peuvent communiquer au-delà du réseau local, et les autres serveurs DNS sont configurés avec l'adresse du redirecteur.

Le redirecteur se comporte comme un gardien qui canaliserait toutes les requêtes venant de l'extérieur. Il est possible d'installer un logiciel pare-feu ou d'autres mesures de protection sur le redirecteur seulement, et non pas sur tous les serveurs DNS d'une organisation. Il faut noter aussi que le rôle de redirecteur est associé au serveur dans son ensemble, et non pas à une ou plusieurs zones particulières.

Quand un redirecteur reçoit une requête de résolution de nom, il accède à des ressources externes et renvoie la réponse au serveur DNS qui a transmis la requête de départ.

Si le redirecteur ne peut pas répondre à la requête, le serveur DNS d'origine peut tenter de la résoudre lui-même.

Comme cela peut aller à l'encontre du but recherché en désignant un serveur comme redirecteur, une option supplémentaire peut être activée pour configurer les serveurs DNS interne comme serveurs esclaves. Ces « esclaves »,qui sont des serveurs DNS configurés que des redirecteurs, renvoient un message d'erreur si le redirecteur ne peut pas résoudre la requête. Un serveur esclave ne tente pas de contacter d'autres serveurs DNS si son redirecteur désigné n'a pas pu traiter la requête .En d'autres termes, un esclave se contente d'émettre à son redirecteur une requête récursive.

CHAPITRE II : IMPLEMENTATION DU DNS

II.1. La structure des fichiers DNS

Les serveurs de noms non Microsoft font en général recours à l'édition manuelle de fichiers de texte pour créer les fichiers de zone formant l'espace de noms de domaines. Ces fichiers doivent être créés suivant une syntaxe spécifique que DNS peut lire. Le serveur DNS Microsoft comprend un gestionnaire DNS, une interface graphique utilisateur qui permet d'afficher les paramètres des fichiers de zone et d'y insérer des entrées à travers cette interface et non dans les fichiers eux-mêmes. DNS permet aussi la gestion de plusieurs serveurs DNS à partir d'un même emplacement.

Même s'il est possible d'utiliser le gestionnaire DNS pour créer ou modifier des fichiers de zones sans avoir à en connaître la syntaxe, il est essentiel de comprendre le contenu et la structure de ces fichiers. En fait, le gestionnaire DNS fait référence à de nombreux enregistrements du fichier de zone à l'aide de leur nom syntaxique. Ainsi, que vous utilisiez un éditeur de texte ou le gestionnaire DNS pour la création et la modification des fichiers de zone, l'utilité des divers enregistrements doit être connue.

Un fichier de zone contient les enregistrements de ressources correspondant à la portion du domaine couverte par cette zone. Dans la structure de Windows NT, ces fichiers sont stockés dans le répertoire C:\WIN NT\ SYSTEM 32\ dns pour une installation par défaut.

Un serveurs DNS peut utiliser trois types de fichiers: des fichiers de zone, un fichier cache et un fichier de recherche inverse. Il peut y avoir aussi un fichier de démarrage servant à initialiser le serveur DNS.

Cependant, un serveur DNS Microsoft est habituellement démarré à partir des valeurs stockées dans la base de registres.
L'initialisation à partir d'un fichier de démarrage est incluse dans DNS Microsoft à des fins de compatibilité avec les fichiers des serveurs DNS basés sur BIND.

II.1.1. Le fichier de zone

Les fichiers de zone comprennent l'extension .dns. Il y' a dans le répertoire **dns \ samples**, un fichier nommé **place.dns** qui est un exemple de fichier de zone qu'il est possible d'éditer manuellement et d'utiliser. Il est, bien entendu, possible d'utiliser le gestionnaire DNS et son interface graphique pour créer des fichiers de zone et les enregistrements qu'ils contiennent.

Le gestionnaire DNS peut être utilisé pour créer des entrées de zone, même sans connaissance de la syntaxe des enregistrements. Il crée cependant les entrées dans les fichiers de zone, respectant la syntaxe correcte, à des fins de compatibilité avec d'autres serveurs. Il est important de connaître la raison de chaque enregistrement et la signification des paramètres.

Voici les différents enregistrements que l'on retrouve dans un fichier de zone:

a) ***Enregistrement SOA***

L'enregistrement SOA (Start Of Autority) littéralement début d'autorité, indique le point de départ des informations d'une zone, c'est-à-dire le serveur principal d'une zone. Ce serveur, désigné par l'enregistrement SOA, fait autorité pour la
zone dans le cadre de la résolution de noms de la zone.
Il est aussi le seul à pouvoir recevoir des modifications de la zone.

Cet enregistrement contient également d'autres informations :

- *Hôte source*: c'est le nom de l'hôte qui dispose de la copie en lecture/ écriture du fichier de zone.
- *Contact e-mail*: il s'agit de l'adresse e-mail de la personne qui maintient le fichier. Elle doit comporter un point au lieu du symbole @ habituel.
- *Numéro de série*: c'est le numéro de version du fichier de zone. Ce numéro doit changer à chaque modification du fichier; il varie automatiquement lorsqu'on utilise le gestionnaire DNS pour modifier le fichier de zone.

- *Intervalle de rafraîchissement*: c'est le délai, exprimé en secondes, qui doit être observé par un serveur secondaire avant de demander au serveur maître si des modifications se sont produites dans le fichier de la base de données. Si c'est le cas, le serveur secondaire demande un transfert de zone.

- *Intervalle avant nouvelle tentative*: ce paramètre indique le délai en secondes que doit respecter un serveur secondaire avant de réessayer un transfert de zone en cas d'échec.

- *Heure d'expiration*: il s'agit de l'intervalle, exprimé en secondes, pendant lequel un serveur secondaire tente un transfert de zone. Lorsque ce délai est passé, l'ancien contenu de la zone est effacé

- *Durée de vie minimale*: c'est la durée, en secondes, pendant laquelle un serveur peut maintenir dans son cache un enregistrement de la base de données. Ce paramètre est envoyé dans la réponse aux requêtes résolues à partir du fichier de la base de données concerné.

- Un enregistrement individuel de ressource peut contenir une durée de vie qui, dans ce cas écrase cette valeur.

Il ne peut exister qu'un enregistrement SOA dans une zone. Un enregistrement SOA résout un nom de domaine en nom d'hôte.

Sa syntaxe est la suivante:

IN SOA < hôte source > < contact e-mail > < série > < intervalle rafraich> < intervalle tentative > < heure d'expiration > < durée de vie >

L'exemple suivant traduit la syntaxe de cet enregistrement

@ IN SOA ns1.erudite.com knolford.erudite.com. (
; serial number
10800; refresh [3 heures]
3600; retry [1 heure]
604800; expire [7 jours]
86400; time to live [1 jour])

Dans cet exemple, le symbole @ identifie le serveur local, et IN un enregistrement internet. Le nom FQDN du serveur NS1 doit se terminer par un point.
Il faut noter aussi que l'adresse e-mail de l'administrateur comprend un point au lieu du symbole habituel @. De plus, si l'enregistrement SOA occupe plus d'une ligne, la première doit finir par une parenthèse ouvrante, et la dernière par une parenthèse fermante.

b) ***Enregistrement de serveurs de noms (NS)***

L'enregistrement de serveur de noms, enregistrement NS (Name Server), sert à identifier les serveurs de noms faisant également autorité pour une zone. Ils sont donc vus par les autres ordinateurs comme une source d'information autorisée pour la zone apte à répondre aux requêtes portant sur la zone. Contrairement à l'enregistrement SOA, il peut exister plusieurs enregistrements NS dans une zone.

Un enregistrement NS facilite la délégation en identifiant le serveur DNS de chaque zone. Un enregistrement NS est présent dans toutes les zones de recherche directe ou inverse.

La syntaxe de l'enregistrement NS se présente comme suit :

< domaine > IN NS < hôte serveur de noms >

c) Enregistrement d'hôte local

C'est un enregistrement d'hôte standard, mais qui contient un nom d'hôte spécial ainsi que l'adresse de bouclage IP normale. Cette adresse redirige ou «boucle » le trafic TCP/IP vers l'hôte qui génère le trafic. Ainsi l'enregistrement suivant affecte le nom d'hôte localhost à l'adresse de bouclage 127.0.0.1:

Localhost IN A 127.0.0.1

Cet enregistrement permet à un client d'interroger localhost et de recevoir l'adresse de bouclage normale.

d) Enregistrement d'hôte

Cet enregistrement spécifie l'adresse IP d'un hôte donné. Tous les hôtes qui disposent d'une adresse IP statique doivent avoir une entrée dans cette base de données. Les clients qui ont une adresse dynamique sont résolus avec d'autres méthodes, à l'aide d'un serveur sous Linux, par exemple.
La plupart des entrées d'un fichier de base de données sont des enregistrements d'hôtes.
Sa syntaxe est la suivante:

< Nom d'hôte > IN A < adresse IP de l'hôte >

e) Enregistrement CNAME

L'enregistrement de nom canonique (CNAME, Canonical Name) permet de définir un alias de manière à avoir plus d'un nom associé à une adresse IP.
Sa syntaxe est la suivante:

< nom d'alias > CNAME < nom d'hôte >

A l'aide d'enregistrements CNAME, il est possible de combiner un serveur FTP et un serveur web sur le même hôte.

L'exemple suivant associe le serveur nommé InetServer à une adresse TCP/IP, puis crée deux alias pour ce serveur:

```
InetServer      IN      A       136.107.3.43
FTP CNAME   InetServer
www   CNAME  InetServer
```

Ces enregistrements montrent comment il est simple de modifier le serveur sur lequel sont fournis les services, tout en continuant à permettre aux clients qui utilisent le nom original d'accéder au nouveau serveur. Ainsi, si le serveur web est déplacé sur une autre machine nommé NewInet, le fichier de zone sera modifié comme suit:

```
InetServer   IN   A   136.107.3.43
FTP CNAME   InetServer
NewInet   IN   A   136.107.1.107
WWW    CNAME   NewInet
```

Le seul changement nécessaire ici pour donner l'accès au nouveau serveur a consisté à modifier des entrées dans le serveur DNS. Aucune modification n'est nécessaire sur les clients.

f) Enregistrement de serveurs de courrier

L'enregistrement de serveur de courrier (MX, Mail exchange) indique le nom de l'hôte qui est en charge de la gestion du courrier pour ce domaine. Si plusieurs serveurs de courrier sont spécifiés, un paramètre de préférence spécifie dans quel ordre ils doivent être utilisés. Si le premier serveur ne répond pas, le second est contacté, et ainsi de suite.

Sa syntaxe se présente de cette manière:

< Domaine > IN MX < Préférence > < hôte serveur de courrier >

g) Enregistrement PTR

Le fichier de recherche inverse a des entrées qui permettent de résoudre des adresses IP en noms d'hôtes. C'est normalement DNS qui est utilisé pour résoudre un nom d'hôte en adresse IP, ce qui explique que le processus opposé s'appelle recherche inverse. Les fichiers sont nommés en fonction de la classe du réseau, mais les octets sont inversés par rapport à l'ordre normal.

Les enregistrements PTR (pointer) constituent les entrées de recherche inversée. Ils spécifient l'adresse IP en ordre inverse à la manière des noms DNS, avec l'information la plus spécifique en tête et le nom d'hôte correspondant.
La syntaxe d'un enregistrement PTR est la suivante:

<nom domaine inverse ip > IN PTR < nom d'hôte >

Après cette brève description des enregistrements rencontrés dans le fichier de zone nous allons décrire comment se fait l'initialisation des données du DNS.

II.2. L'initialisation des données du DNS

La première étape dans l'initialisation des serveurs de noms consiste à convertir la table d'hôtes en son équivalent pour le DNS. La version de la table pour le DNS est composée de plusieurs fichiers. L'un d'eux relie les noms d'hôtes à leur adresse, d'autres relient les adresses à leur nom d'hôte. La correspondance nom-adresse est parfois appelée correspondance directe (forward mapping) et la correspondance adresse-nom est parfois appelée correspondance inverse (reverse mapping). Chaque réseau a son propre fichier de correspondance inverse.

La section suivante décrit comment se fait le démarrage d'un serveur primaire.

II.3. Le démarrage d'un serveur primaire

Une fois les fichiers de zone créés, vous êtes prêt à démarrer un couple de serveurs de noms. Vous aurez besoin de démarrer deux serveurs: un serveur primaire et un serveur-esclave. Avant de démarrer un serveur, il convient de s'assurer que le démon (type de programme informatique ou un processus s'exécutant en arrière-plan plutôt que sous le contôle direct d'un utilisateur) syslog fonctionne.

De cette manière, si le serveur de noms détecte une erreur lors de la lecture du fichier de configuration ou des fichiers de zone, il transmettra un message à syslog. Si l'erreur est critique, le programme de serveurs de noms s'arrêtera.
Pour démarrer le serveur de noms, il est crucial de prendre l'identité de root, car le serveur se met à l'écoute sur un port réservé qui requiert ce privilège.
Aucun privilège n'est nécessaire par la suite. La première fois, démarrer le serveur à partir de la ligne de commande, afin de tester facilement son fonctionnement.

II.4. Le démarrage d'un serveur-esclave

Pour la robustesse, il est nécessaire d'initialiser un autre serveur de noms. Vous pouvez initialiser plus de deux serveurs de noms, deux étant un minimum.
Si vous avez un seul serveur et qu'il vient à s'arrêter, plus personne ne peut effectuer de recherche. Un second serveur partage la charge avec le premier ou la reçoit en totalité si le premier est arrêté. Il est possible d'initialiser un autre serveur primaire, mais cela est déconseillé; il vaut mieux démarrer un serveur-esclave.

Mais il demeure toujours possible de le transformer en serveur primaire, si vous avez du temps à consacrer à l'exploitation de plusieurs serveurs primaires.
Un serveur sait s'il est serveur primaire ou esclave d'une zone grâce au fichier **named.conf.**
L'enregistrement NS n'indique pas qui est le serveur primaire ou qui est le serveur-esclave d'une zone, il indique seulement qui sont les serveurs.

La différence fondamentale entre un serveur primaire et un serveur-esclave est l'origine des données. Un serveur primaire obtient les données à partir de fichiers, alors qu'un esclave les obtient d'un autre serveur de noms, via le réseau. Ce dernier processus s'appelle un transfert de zone.

Un serveur-esclave n'est pas obligé d'obtenir ses données d'un serveur primaire;
il peut les obtenir d'un autre serveur-esclave.

Un serveur-esclave présente l'avantage de n'avoir à gérer qu'un seul jeu de données, celui situé sur le serveur primaire. Il est inutile de vous soucier de la synchronisation entre les serveurs de noms; les esclaves le font pour vous. Toutefois, un esclave ne se resynchronise pas instantanément; il teste régulièrement sa validité. L'intervalle de scrutation est l'une des valeurs de l'enregistrement SOA.

Un serveur-esclave n'a pas besoin de télécharger la totalité des fichiers de zone depuis un autre serveur. Les fichiers db.cache et db.127.0.0 sont les mêmes sur un serveur primaire et sur un esclave. Il suffit d'en garder une copie sur l'esclave, ce qui signifie que pour 0.0.127.in-addr.arpa, un esclave est serveur primaire.

Ce bref survole des théories relatives à l'implémentation d'un système de noms de domaine nous achemine à celles relatives à l'exploitation et maintenance du DNS.

CHAPITRE III : EXPLOITATION ET MAINTENANCE DU DNS

Après l'implémentation du système de Noms de Domaine, la tâche la plus cruciale qui suit est son exploitation.

Par exploitation nous comprenons la gestion des différents sous domaines, la délégation d'autorité, la prévention des pannes éventuelles, l'expansion du système et le traitement des pannes le cas échéant. Ce chapitre fera le contour de ces différents points.

III.1. L'exploitation du DNS

III.1.1. L'expansion de domaine : le choix du nombre de serveurs

Pour gérer un nom de domaine, nous avons besoin d'au moins deux serveurs de noms : le serveur maître et le serveur esclave.

Cependant deux serveurs de noms ne suffisent généralement pas pour gérer un réseau de grande taille. Ainsi, il arrive parfois de gérer quatre serveurs ou plus dont certains peuvent être sur des sites distants. Le choix du nombre de serveurs incombe à chaque domaine.

Pour mener à bien la tâche de gestion de domaine, il convient d' :

- installer au moins un serveur de noms par réseau ou sous-réseau afin de s'affranchir des pannes de routeur ;
- installer un serveur de noms sur les serveurs de stations sans disque, pour servir spécifiquement à ces dernières ;

- installer les serveurs de noms à proximité des machines multi-utilisateurs importantes qui génèrent à priori de nombreuses requêtes.
  Il faut toutefois évaluer les risques d'avoir un serveur de noms accessible à de nombreuses personnes, donc une ressource critique du point de vue de la sécurité ;

- installer au moins un serveur de noms sur un site distant, pour rendre les données accessibles même si le réseau local est hors-service. Il peut sembler inutile de chercher à garantir le service de noms dans le cas où le réseau local est inaccessible; en fait, le serveur distant est surtout utile lorsque le réseau local est accessible mais qu'aucun serveur de noms interne n'est opérationnel.

III.1.2. L'ajout des serveurs

Après avoir effectué le choix du nombre de serveurs en fonction de la taille du réseau, l'étape suivante est leur ajout dans le réseau.

Pour ajouter de nouveaux serveurs de noms, le plus simple est de créer des serveurs-esclaves. Toutefois, l'ajout inconsidéré de serveurs-esclaves peut conduire à des problèmes.

En effet, si les serveurs-esclaves d'une zone sont trop nombreux, le serveur de noms primaire peut se retrouver ponctuellement surchargé par les tests de synchronisation issus des serveurs-esclaves.

Plusieurs moyens permettent de résoudre ce genre de problème:

- Création de plusieurs maîtres primaires.
- Pour certains esclaves, chargement de leur zone à partir d'autres esclaves.
- Création de serveurs cache.
- Création d'esclaves « partiels »

III.1.3. La gestion des sous-domaines

Lorsqu'un domaine devient trop grand ou lorsque l'on décide de répartir sa gestion entre plusieurs entités, il est nécessaire de le diviser en sous-domaines. Ceux-ci deviennent les enfants du domaine courant dans l'espace de noms, le domaine courant étant le parent. La responsabilité des sous-domaines étant déléguée, chaque sous-domaine dispose de sa propre zone différente de la zone parent.

Une bonne gestion des sous-domaines comprend le découpage du domaine, le choix du nom des sous-domaines et la délégation à ces sous-domaines pour créer de nouvelles zones. Comme tout parent responsable, le domaine parent s'assure en permanence de la pérennité des liens entre sa zone et celles de ses enfants.

En raison de l'importance du service de noms pour la navigation entre sites, cette bonne gestion est fondamentale pour le fonctionnement correct du réseau. Une délégation incorrecte vers des serveurs de noms peut conduire à l'inaccessibilité d'un site alors que la perte de la connectivité vers les serveurs de la zone parente peut empêcher les hôtes d'un site de contacter tout hôte situé à l'extérieur de la zone.

Dans cette section, il s'agit de présenter un point de vue sur la création de sous-domaines et de décrire en détail les moyens d'y parvenir. Il sera question de la gestion des relations entre parent et enfants et de la gestion du processus de découpage d'un grand domaine en petits sous-domaines avec un minimum d'interruption et de perturbation.

III.1.3.1. Noms de sous-domaines

Il faut maintenant choisir les noms des sous-domaines, dans la mesure du possible, en collaboration avec l'administrateur du domaine et les membres du futur sous-domaine.

Le choix du nom peut être une source de problèmes. Il est conseillé d'utiliser un schéma de noms cohérent entre les sous-domaines, afin de faciliter le repérage de la part des utilisateurs et la mémorisation des noms.

En laissant le champ totalement libre à l'administrateur du sous-domaine, on peut arriver à un espace de noms chaotique dans la mesure où certains administrateurs voudront utiliser des noms géographiques, d'autres des noms liés à leur organisme. Certains voudront des noms abrégés, d'autres des noms complets. C'est cette diversité dans le choix de noms de sous-domaines par différents administrateurs qui motive le choix du nom de sous-domaine en collaboration avec l'administrateur du domaine pour garantir la cohérence dans le réseau.

Ainsi, dans certaines sociétés dynamiques le nom de département change souvent, en fonction des projets. Un nom de sous-domaine basé sur ces noms serait une erreur. Il vaut mieux utiliser des noms non significatifs et stables dans le temps.

Dans cette optique les noms géographiques sont plus stables que ceux des organismes, mais ils peuvent paraître obscurs en dehors de l'entreprise.

De plus, Il ne faut pas utiliser des noms existants ou réservés aux domaines de niveau supérieur pour un nom de sous-domaine. Il peut sembler intéressant d'utiliser des abréviations géographiques en deux lettres pour des sous-domaines internationaux ou d'utiliser des noms de domaine de niveau supérieur, tel que net pour un département réseau (network), mais cela peut poser des problèmes.

En nommant com le sous-domaine du département communication, il y a risque d'avoir des difficultés pour communiquer avec les hôtes du domaine de niveau supérieur com.

Aussi, il ne faut pas sacrifier la lisibilité à l'aspect pratique. Des noms sur deux lettres sont rapides à taper, mais impossibles à reconnaître.

III.1.3.2. Nombre de sous-domaines

Après avoir décidé de créer des sous-domaines, il faut choisir leur position dans l'espace de noms. Si une société est organisée, en quatre secteurs d'activité, on peut choisir de créer quatre sous-domaines, un par secteur.

On peut aussi créer de nombreux petits sous-domaines que ne créer que quelques grands sous-domaines. Tout est affaire de compromis.

Une délégation comptant un petit nombre de grands sous-domaines n'engendre que peu de travail au domaine parent car il y' a peu de délégation à surveiller.

Mais les grands domaines ont besoin de serveurs plus rapides et dotés de plus de mémoire. De plus, leur gestion n'est pas distribuée.

En outre, en mettant en oeuvre des sous-domaines au niveau d'un site, même les groupes autonomes devront se partager une zone unique et devront être gérés par une seule entité.

La délégation à un grand nombre de petits sous-domaines peut être une casse-tête pour l'administrateur du domaine parent. Pour tenir à jour les informations de délégation, il faut contrôler en permanence l'identité des hôtes qui hébergent les serveurs de noms ainsi que l'identité des zones sur les quelles ils font autorité. Les informations évoluent à chaque ajout d'un serveur de noms dans un sous-domaine ou lorsque l'adresse d'un serveur est modifiée.

Si tous les sous-domaines sont gérés par des personnes différentes, cela nécessite un grand nombre de prises de contact et donc une surcharge de travail.

D'un autre côté, les sous-domaines sont plus petits et plus faciles à gérer et les administrateurs des zones sont plus proches de leurs utilisateurs.

Au vu de ces avantages et inconvénients, il peut sembler difficile de faire un choix. Le mieux est probablement de suivre l'organisation naturelle de l'entreprise. Certaines sociétés gèrent les machines et les réseaux au niveau du site; d'autres ont des groupes de travail décentralisés et relativement autonomes qui gèrent tout eux-mêmes.

La structure d'un domaine devrait suivre celle de l'entreprise, entre autres en raison *de* son organisation technique. Si une entité exploite des réseaux, attribue des adresses IP et gère des hôtes, cette entité doit aussi gérer son sous-domaine.

Si l'administrateur n'a pas l'adhésion de tout le monde sur la façon d'organiser l'espace de noms, il doit tenter de fixer des règles précisant à partir de quel moment un groupe peut obtenir son propre sous-domaine.

III.1.3.3. La création des sous domaines

La création de sous-domaines est guidée par :

- le besoin de répartir la gestion d'un domaine entre plusieurs organismes;
- les difficultés liées à la grande taille du domaine, le fractionnement permettant de faciliter la gestion et de réduire la charge des serveurs de noms faisant autorité;

- la nécessité de distinguer l'appartenance des hôtes en les plaçant dans des sous-
- domaines spécifiques.

III.2. La maintenance du DNS

Comme plusieurs autres applications Internet, le DNS présente des vulnérabilités qui sont souvent utilisées pour perpétrer des actes malveillants. Ces vulnérabilités se trouvent à plusieurs niveaux du flux de données DNS :

- dans les procédures de mise à jour dynamiques du fichier de zone d'un serveur DNS primaire, des mises à jours non autorisées vont affecter l'intégrité des données de ce fichier ;

- dans le processus de transfert de zone entre le serveur primaire et les serveurs secondaires, une usurpation de l'identité du serveur primaire va avoir comme conséquence l'utilisation de données corrompues au niveau des serveurs secondaires ;

- dans les processus de requêtes DNS des clients, les résolveurs peuvent délivrer des réponses fausses en cas d'usurpation d'identité du serveur cache ;

- dans les échanges entre serveurs autoritaires et serveurs cache, le trafic peut être corrompu de telle sorte que le serveur cache soit « pollué » avec des informations fausses .

Le schéma ci-dessous traduit l'origine des différentes pannes qui hantent le fonctionnement du DNS.

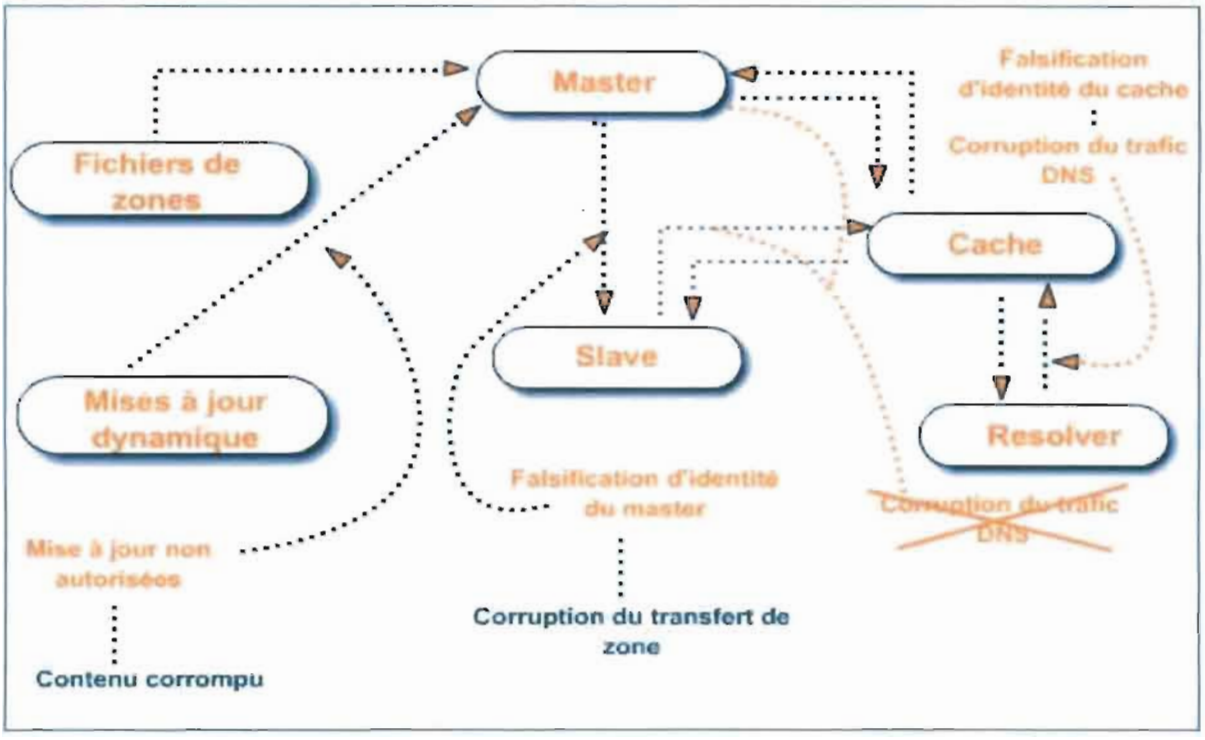


Figure 6 : Schéma de bilan des vulnérabilités résolues par DNSSec

III.2.2. Anticipation de pannes

Un réseau finit toujours par tomber en panne: défaillances matérielles, bogues dans les logiciels, erreurs humaines. Les conséquences peuvent être mineures comme la perte de connexion pour quelques utilisateurs, mais elles sont parfois catastrophiques en guise d'exemple la perte de données importantes ou de travaux fondamentaux.

Puisque le DNS s'appuie fortement sur les réseaux, il est particulièrement vulnérable aux ruptures de connexion. Heureusement, sa conception prend en compte les imperfections du réseau.

Ceci est d'autant plus critique que le protocole DNS a une fonctionnalité qui peut constituer une faille de sécurité, notamment le fait pour un serveur DNS d'envoyer des informations autres que celles demandées dans les requêtes.

On comprend donc aisément que la sécurité des échanges de messages DNS tant au niveau de l'authentification que de l'intégrité des données revêt un intérêt capital.

Les solutions proposées pour apporter des solutions aux problèmes de sécurité liés au DNS font appel à des techniques cryptographiques et se situent à deux niveaux :

- d'abord au niveau des transactions entre clients et serveurs DNS (en général entre un résolveur et un serveur cache) ainsi qu'au niveau des échanges entre serveurs primaires et serveurs secondaires pour les transferts de zones ;
- ensuite au niveau de la validation des données fournies dans les réponses aux requêtes DNS afin d'assurer qu'elles ne souffrent d'aucun défaut d'intégrité.

III.2.1. La sécurisation des transactions avec TSIG

TSIG est un mécanisme qui permet de sécuriser les transactions entre serveurs **DNS** par l'utilisation de signatures électroniques. **TSIG** est essentiellement utilisé pour la communication entre serveurs primaires et leurs serveurs secondaires dans les processus de transfert de zone, mais peut aussi être mis en oeuvre pour les mises à jour dynamiques des fichiers de zones et les transactions entre résolveurs et serveurs cache.

TSIG utilise une technique de cryptographie à clé symétrique, c'est-à-dire que les parties impliquées dans la communication utilisent un secret partagé qui est valable aussi bien pour le codage que pour le décodage.

Avec **TSIG**, tous les messages **DNS** sont signés ; ceci concerne aussi bien les requêtes que les réponses à ces dernières. Il est important de noter que **TSIG** permet uniquement d'authentifier la source des messages **DNS** (requêtes et réponses); il ne permet pas de vérifier la validité des données **DNS** incluses dans le message **DNS** qui peuvent provenir, elles, d'un serveur autoritaire autre que celui qui fournit la réponse. Ceci est du au fait que **TSIG** permet de signer les messages **DNS**, et non les **RR** contenus dans ces derniers.

Pour éviter une interception de signature et son usage malveillant ultérieur, les signatures **TSIG** ne sont utilisées qu'une fois et les messages signés contiennent tous une information relative à la date et l'heure de création de la signature. Ceci nécessite par conséquent une synchronisation **NTP** entre les parties qui échangent les messages **DNS**.

Il convient aussi de signaler qu'il faut générer autant de clés (secrets partagés) que de couples de machines impliqués dans la communication sécurisée.

Le schéma ci-dessous montre le bilan des vulnérabilités résolues et non résolues par l'utilisation de **TSIG** ; il en ressort que le problème de la possible corruption du trafic entre les serveurs autoritaires et les serveurs cache reste entier.

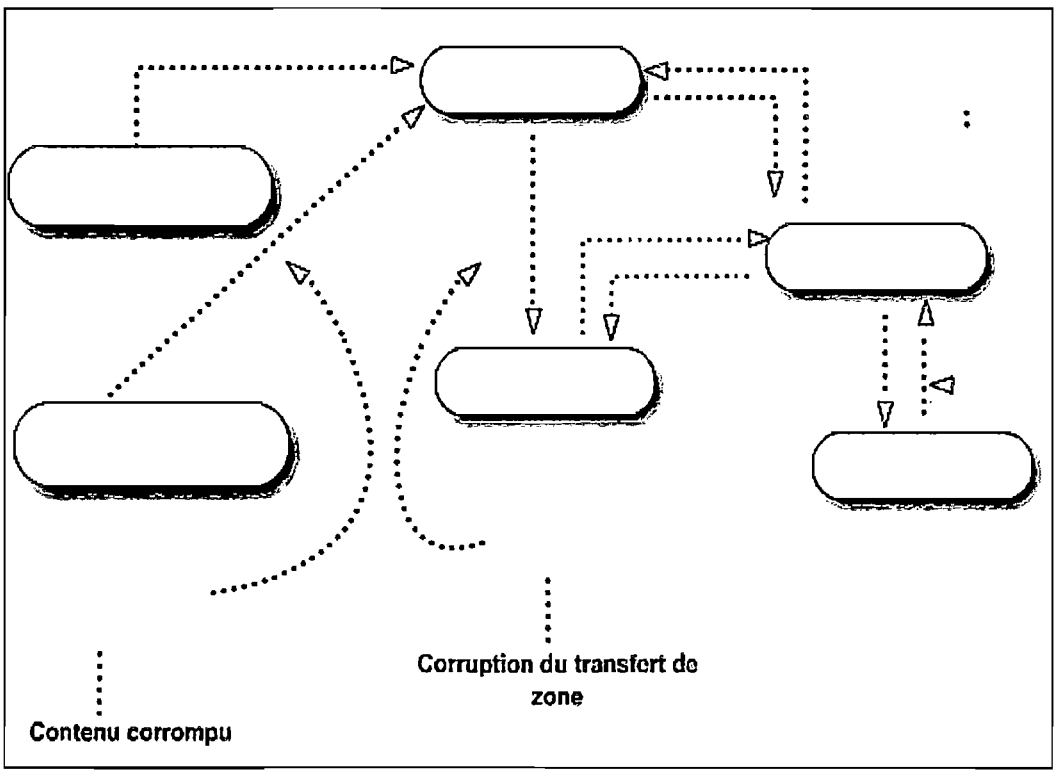


Figure 5: Schéma des vulnérabilités résolues et non résolues par TSIG

Pour sa mise en oeuvre dans le cadre des procédures de transfert de zone, TSIG requiert essentiellement l'exécution des taches suivantes :

- La confection de la clé correspondant au secret partagé ;
- la confection au niveau du serveur primaire d'une liste d'accès spécifiant les clés donnant accès à la fonction de transfert de zone ;
- la spécification au niveau du serveur secondaire, de la clé à utiliser pour un transfert de zone à partir du serveur primaire ;

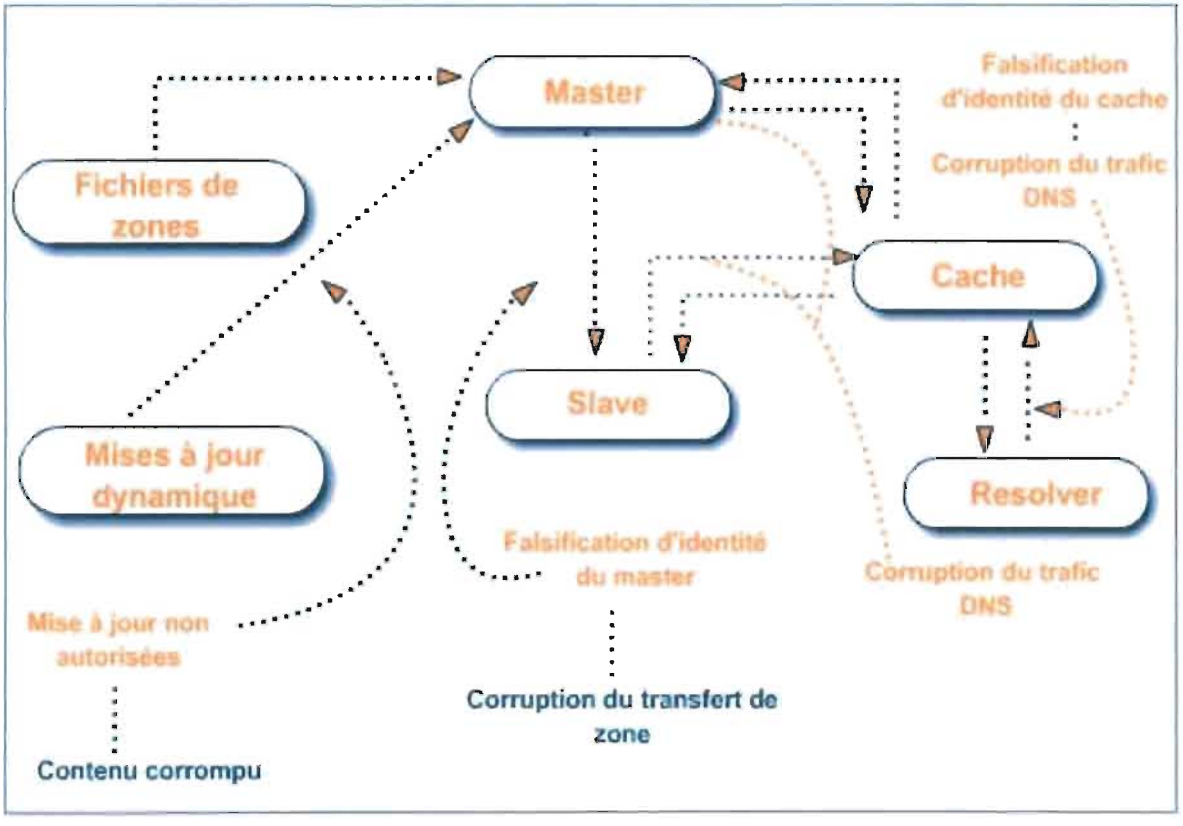


Figure 4 : Schéma des différentes pannes du DNS

Généralement, les attaques malveillantes sur le service DNS ont comme objectif premier la re-direction de trafic vers d'autres serveurs à partir desquels d'autres attaques sont perpétrées en vue du vol de données d'authentification par exemple.

Le DNS est au coeur du fonctionnement de l'Internet, et les informations échangées sont en principe des informations publiques ; le problème auquel on est confronté sur le plan de la sécurité est de savoir si l'information qu'on souhaite avoir provient de la source habilitée à la donner et si cette information n'a pas été modifiée par une tierce partie.

- la synchronisation des horloges des serveurs primaire et secondaire.

a) Confection de la clé

Une manière simple de générer la clé pour une paire de serveurs consiste à utiliser l'utilitaire ***dnssec-keygen*** fourni avec **BIND 9**.

 # dnssec-keygen -a HMAC-MD5 -b 128 -n HOST host1-host2

Cette commande va générer une clé d'une longueur de 128 bits codée en base-64 avec l'algorithme **HMAC-MD5** ; en réalité, la spécification du type n'est pas utile pour la génération d'un secret partagé. Le nom de la clé sera **host1-host2**.

La clé est en fait stockée dans deux fichiers créés par la commande et qui ont pratiquement le même contenu.

La clé générée ne doit en aucun cas apparaître dans le fichier de zone ; elle doit être plutôt enregistrée au niveau des deux serveurs **host1** et **host2** dans un fichier avec des droits d'accès appropriés qui sera pris en considération au démarrage de **BIND** à travers une l **include** dans le fichier **named.conf**. Le fichier en question, qu'on pourrait nommer **host1-host2.key**, aura comme contenu :

```
Key host1-host2 {
    algorithm hmac-md5;
    secret "abcd1234...";
} ;
```

- la synchronisation des horloges des serveurs primaire et secondaire.

a) Confection de la clé

Une manière simple de générer la clé pour une paire de serveurs consiste à utiliser l'utilitaire ***dnssec-keygen*** fourni avec **BIND 9**.

 # dnssec-keygen -a HMAC-MD5 -b 128 -n HOST host1-host2

Cette commande va générer une clé d'une longueur de 128 bits codée en base-64 avec l'algorithme **HMAC-MD5** ; en réalité, la spécification du type n'est pas utile pour la génération d'un secret partagé. Le nom de la clé sera **host1-host2**.

La clé est en fait stockée dans deux fichiers créés par la commande et qui ont pratiquement le même contenu.

La clé générée ne doit en aucun cas apparaître dans le fichier de zone ; elle doit être plutôt enregistrée au niveau des deux serveurs **host1** et **host2** dans un fichier avec des droits d'accès appropriés qui sera pris en considération au démarrage de **BIND** à travers une l **include** dans le fichier **named.conf**. Le fichier en question, qu'on pourrait nommer **host1-host2.key**, aura comme contenu :

```
Key host1-host2 {
    algorithm hmac-md5;
    secret "abcd1234...";
} ;
```

abcd1234... étant la clé générée par la commande **dnssec-keygen.**
Dans le fichier **named.conf**, on ajouterait donc la ligne suivante :

include host1-host2.key

b) La configuration TSIG au niveau du serveur primaire

Dans le fichier de configuration du serveur, on doit spécifier la clé à fournir par les serveurs secondaires pour être autorisés à effectuer un transfert de zone. Si nous considérons que la zone en question est **universiteub.bi** et que le fichier de zone se nomme **universiteub.bi.hosts**, alors le fichier de configuration named.conf du serveur primaire contiendrait entre autres les lignes suivantes :

```
key host1-host2 {
    algorithm hmac-md5;
    secret "abcd1234...";
};

zone «universiteub.bi » {
        type master ;
        file « universiteub.bi.hosts » ;
        allow-transfer { key host1-host2; } ;
}
```

Les deux premières lignes de cet extrait (**key host1-host2 { algorithm hmac-md5 ;)** de **named.conf** spécifient la clé à utiliser pour une communication sécurisée avec le serveur primaire de la **zone universiteub.bi.**

c) La synchronisation des horloges

La référence temporelle est fondamentale pour un système DNS sécurisé; aussi bien **TSIG** que **DNSSec** font appel à de telles références (signature datée pour **TSIG** et définition de la période de validité des enregistrements signés pour **DNSSec**). Pour une synchronisation universelle, indépendante des fuseaux horaires, on utilisera systématiquement le temps universel (**TU**) ; il est recommandé de réaliser la synchronisation en faisant appel à des serveurs **NTP** (Network Time Protocol). **SSec** représente une série d'extensions apportées au protocole DNS pour sécuriser le trafic **DNS**. Il prend en compte trois aspects de ce trafic : la distribution de clés, l'authentification et l'intégrité des données du fichier de zone. On va donc ici plus loin que **TSIG** dans la mesure où on veille à l'intégrité des données contenues dans les messages échangés et pas uniquement à celle des messages dans leur globalité.

Il faut cependant noter que **TSIG** ne peut être considéré comme superflu avec la mise en oeuvre **DNSSec** avec les nouveaux **RR**, mais en constitue plutôt un utile complément.

Contrairement à **TSIG**, **DNSSec** fait appel à la cryptographie à clés asymétriques. Les systèmes cryptographiques asymétriques font intervenir des paires de clés dont l'une est publique et l'autre privée. La clé publique est diffusée ou fournie de manière ponctuelle. Elle permet de coder des données qui ne pourront être décodées qu'avec la clé privée correspondante qui, elle, est secrète. Par ailleurs, cette même clé privée permet à son détenteur de coder des données dont le décodage avec la clé correspondante qui, elle, est publique permet d'authentifier l'expéditeur du message associé : c'est le principe de la signature électronique. **DNSSec** utilise cette deuxième fonctionnalité de la cryptographie à clés asymétriques.

Le problème avec les systèmes de cryptographie à clé publique est que cette technique requiert beaucoup plus de ressources que les systèmes de cryptographie symétriques. Il n'est donc pas opportun, compte tenu de cette charge de calcul, de procéder au cryptage des fichiers de zone. Par contre, chaque **RRset** (jeu d'enregistrements de ressource de même type se rapportant à la même machine) est signé avec la clé privée de la zone.

La clé publique, elle, est incluse dans les données de zone, ce qui permet de procéder à l'authentification et à la vérification de l'intégrité des données signées.

Cette fonctionnalité est mise en oeuvre à l'aide des nouveaux RR KEY (définition de clés publiques dans le fichier de zone) et **SIG** (spécification des signatures associées aux **RRset**).

Il convient de noter que ce sont les **RRset** qui sont signés, pas les **RR** pris individuellement; un **RRset** peut cependant être constitué d'un seul **RR**. Ceci fait que dans une requête, on ne peut rapatrier uniquement un seul élément d'un **RRset**; c'est tout le **RRset** qui est fourni en réponse aux requêtes, même si l'information demandée n'est contenue que dans un seul élément du **RRset**.

Outre **KEY** et **SIG**, deux autres **RR** ont été introduits : **NXT** pour les réponses négatives à des requêtes et **DS** pour la délégation de signature des zones filles.

d) L'enregistrement de ressources "KEY"

Le **RR KEY** contient la clé publique pour une zone ou une machine. Pour une zone donnée, chaque **RRset** est signé avec la clé privée de la zone.

e) L'enregistrement de ressources "SIG"

Comme indiqué plus haut le **RR SIG** est utilisé pour signer les **RRset.** Ces derniers sont signés avec la clé privée associée à la clé publique spécifiée dans le **RR KEY**. Il convient de préciser que les données signées sont celles obtenues par concaténation du contenu des enregistrements du **RRset** et de tous les champs composant l'enregistrement **SIG**, à l'exception de la signature elle-même. En outre, la signature n'est effectuée qu'après un tri des enregistrements.

f) L'enregistrement de ressources "NXT"

Nous avons vu dans la section précédente que la signature est appliquée sur les données des enregistrements d'un **RRset** et les champs du **RR SIG** à l'exception de la signature. Lorsqu'une réponse à une requête **DNS** est négative (non-existence de l'enregistrement demandé), il est alors nécessaire de trouver un mécanisme d'authentification de cette réponse négative qui est alors enregistrée dans le cache de l'expéditeur de la requête. Le **RR NXT** (NeXT ou Non-eXistenT) sert à authentifier de telles réponses négatives.

La réponse négative à l'aide du **RR NXT** consiste à fournir les deux enregistrements qui seraient respectivement avant et après l'enregistrement recherché, si celui existait.

Ceci explique la nécessité d'ordonner les **RRset** dans le fichier de zone.

En plus de cette information sur les enregistrements qui auraient "encadré" celui recherché, une réponse **NXT** fournit aussi les types d'enregistrement existant pour l'enregistrement précédent.

g) L'enregistrement de ressources "DS"

La vérification de l'authenticité et de l'intégrité des données d'une zone est effectuée à l'aide des clés publiques fournies. Le problème qui peut se poser alors est de savoir si ces clés sont authentiques. La solution à ce problème est la mise en place d'une chaîne de confiance qui consiste à signer les clés publiques des zones par leurs zones parentes :

il s'agit donc en quelque sorte d'une délégation de signature qu' effectue la zone parente au profit de la zone fille pour la signature des données (**RRset**) se trouvant dans cette dernière. Ce processus est réalisé à l'aide du **RR DS** (Delegation Signer).

Le principe est de faire signer par la zone parente la clé publique (**RR KEY**) des zones filles. En d'autres termes, une zone fille sécurisée doit impérativement contenir un **RR KEY** (sa clé publique) signé par la zone parente.

Le schéma ci-dessous donne le bilan des vulnérabilités résolues par **DNSSec**. On remarquera que le trafic **DNS** entre les serveurs autoritaires et les serveurs cache reste encore à sécuriser.

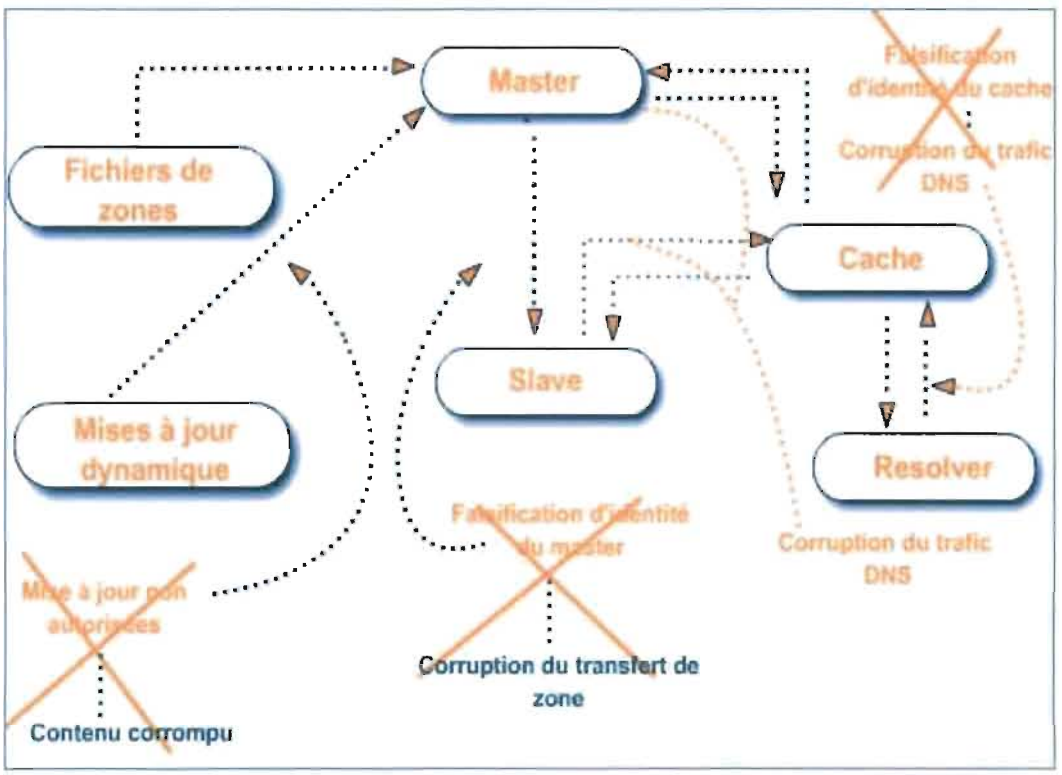


Figure 5: Schéma des vulnérabilités résolues et non résolues par TSIG

Pour sa mise en oeuvre dans le cadre des procédures de transfert de zone, TSIG requiert essentiellement l'exécution des taches suivantes :

- La confection de la clé correspondant au secret partagé ;
- la confection au niveau du serveur primaire d'une liste d'accès spécifiant les clés donnant accès à la fonction de transfert de zone ;
- la spécification au niveau du serveur secondaire, de la clé à utiliser pour un transfert de zone à partir du serveur primaire ;

Il autorise des serveurs de noms multiples et redondants, des redirections de requêtes, des tentatives multiples de chargement de zone, etc. Le DNS ne peut pas se protéger tout seul contre toutes les calamités, mais avec un minimum d'investissement, on peut minimiser l'impact des problèmes.
Les vulnérabilités aux ruptures de connexion peuvent aussi découlées des coupures d'électricité dont les causes peuvent être multiples.

Si tous les hôtes d'un domaine sont arrêtés, le service de noms est inutile. Les problèmes surgissent au retour de l'électricité. Les serveurs de noms sont souvent installés sur de grosses machines de service qui sont parfois les dernières à redémarrer. Les machines à redémarrage rapide doivent donc pouvoir démarrer en l'absence du service de noms.

III.2.3. Le traitement de pannes

Le traitement de panne est une étape qui nécessite la connaissance des différentes pannes qui hantent le réseau.
Les sections suivantes font un relevé de ces pannes et la façon de les réparer.
Une bonne maîtrise des différentes pannes permettrait à maintenir le réseau en fonctionnement.

III.2.3.1. L'interruptions de longue durée (jours)

Avec des interruptions longues, les serveurs peuvent rencontrer des problèmes. S'ils perdent longtemps la connectivité vers les serveurs de la racine, ils cessent de résoudre les requêtes ne concernant pas la zone sur laquelle ils font autorité. Si les serveurs esclaves ne peuvent pas contacter leur serveur-maître, ils finissent même par faire expirer leur propre zone. Pour dépanner, un fichier /etc/hosts temporaire peut-être mis en place. Dans la plupart des cas, il faudra renommer resolv.conf (en resolv.back), arrêter le serveur de noms local (s'il y'en a un) et utiliser/etc/hosts.

On peut aussi transformer temporairement un serveur-esclave qui ne peut pas contacter son maître, en serveur primaire, en modifiant named.conf et en remplaçant le type de serveur (slave) dans la structure zone par le nouveau type (master).Si plus d'un esclave de la même zone sont isolés du reste du monde, on peut en configurer temporairement un en serveur primaire et indiquer aux autres de se synchroniser sur ce nouveau maître.

III.2.3.2. L'interruption de très longue durée (semaines)

Si une très longue interruption isole une zone de l'Internet, il est nécessaire de rétablir artificiellement la connectivité vers les serveurs de la racine. Tout serveur est amené périodiquement à contacter un serveur de la racine. Pour cela, il faut créer temporairement ses propres serveurs de la racine, qu'il faudra supprimer dès le rétablissement de la connexion d'origine. Les serveurs annonçant qu'ils sont serveurs de la racine alors qu'ils ne connaissent pas grand chose sur les domaines de niveau supérieur, sont une des plus grandes abominations de l'Internet. Les serveurs configurés pour interroger une mauvaise liste de serveurs de la racine sont aussi source de graves problèmes.

CHAPITRE IV : IMPLEMENTATION DU DNS A L'UNIVERSITE DU BURUNDI

IV.1. Situation géographique de l'université du Burundi

L'Université du Burundi est implantée sur plusieurs sites séparés par quelques kilomètres les uns des autres. Hormis l'Institut Supérieur d'Agriculture, qui se trouve en Province de Gitega (environ 100 km de Bujumbura), les autres Facultés et Instituts sont localisés à Bujumbura et se répartissent sur six campus: Mutanga, Kiriri, Rohero, CHU-Kamenge, Psychologie-Kamenge et Gihosha- Chaire Unesco.

IV.2. Présentation de l'évolution du réseau informatique de l'Université du Burundi

En 1997, avec l'appui du PNUD, l'Université du Burundi a mis en place la plate-forme de son réseau *Intranet/Internet*, interconnectant dans sa première phase cinq sites : la Bibliothèque centrale, qui abrite le serveur du réseau ; l'administration centrale de l'Université (Rectorat) ; quatre Facultés (Lettres et Sciences humaines, Droit, Sciences et Médecine).

En mai 2004, dans le cadre du Projet « Désenclavement », la *Commission universitaire pour le Développement* (CUD, Coopération belge) a apporté une contribution financière qui a permis de parachever l'interconnexion de tous les services.

L'ensemble Facultés et Instituts, exception faite de l'ISA – Gitega et de l'Institut de Pédagogie appliquée (IPA) au Campus Rohero, sont donc interconnectés à partir du campus Mutanga, qui est le point central d'accès à l'*Internet* et à l'architecture de

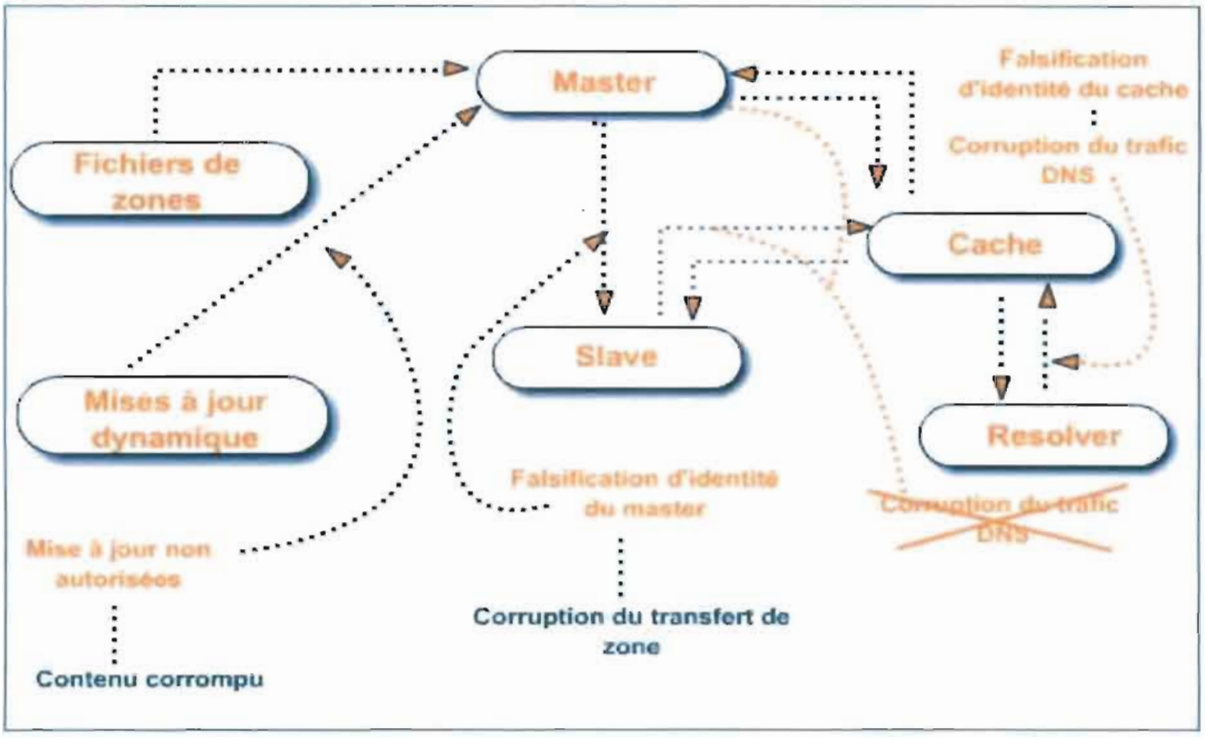


Figure 6 : Schéma de bilan des vulnérabilités résolues par DNSSec

III.2.2. Anticipation de pannes

Un réseau finit toujours par tomber en panne: défaillances matérielles, bogues dans les logiciels, erreurs humaines. Les conséquences peuvent être mineures comme la perte de connexion pour quelques utilisateurs, mais elles sont parfois catastrophiques en guise d'exemple la perte de données importantes ou de travaux fondamentaux.

Puisque le DNS s'appuie fortement sur les réseaux, il est particulièrement vulnérable aux ruptures de connexion. Heureusement, sa conception prend en compte les imperfections du réseau.

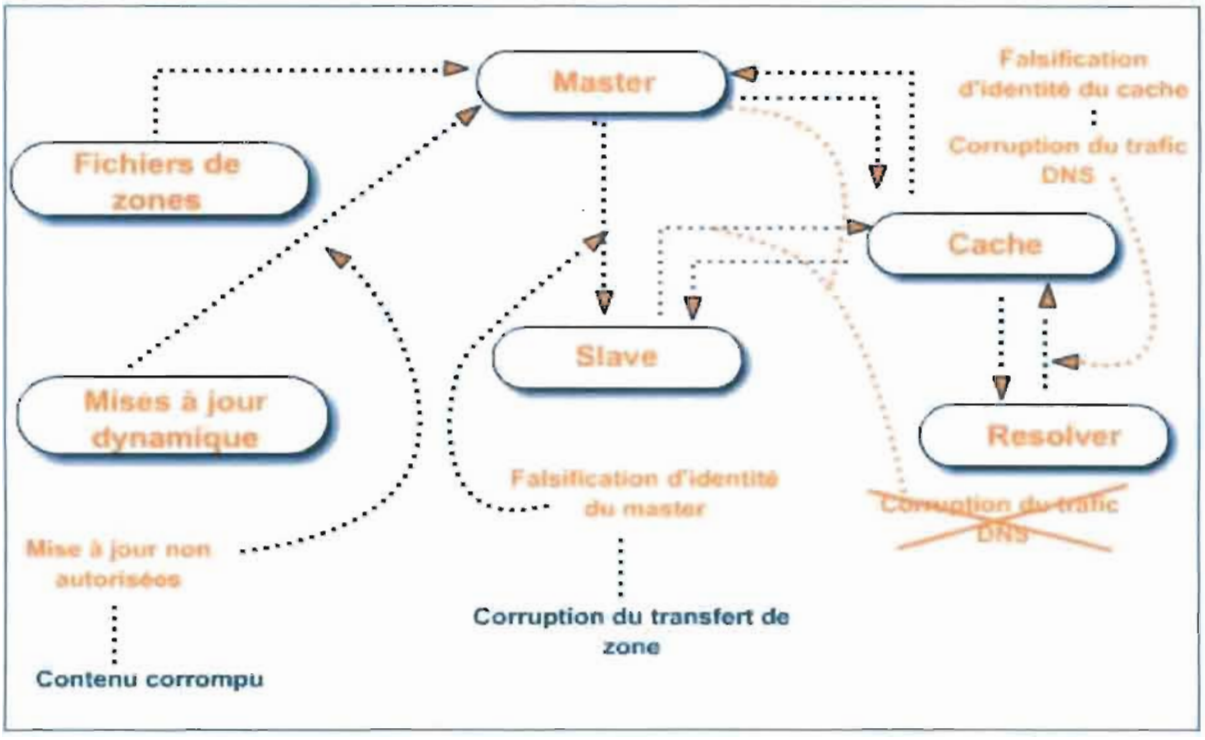


Figure 6 : Schéma de bilan des vulnérabilités résolues par DNSSec

III.2.2. Anticipation de pannes

Un réseau finit toujours par tomber en panne: défaillances matérielles, bogues dans les logiciels, erreurs humaines. Les conséquences peuvent être mineures comme la perte de connexion pour quelques utilisateurs, mais elles sont parfois catastrophiques en guise d'exemple la perte de données importantes ou de travaux fondamentaux.

Puisque le DNS s'appuie fortement sur les réseaux, il est particulièrement vulnérable aux ruptures de connexion. Heureusement, sa conception prend en compte les imperfections du réseau.

La *Coopération Universitaire pour le Développement* (CUD) qui appuie le projet depuis 2004, a fourni le financement nécessaire à travers le volet « Désenclavement de l'Université».

L'image ci-dessous montre l'antenne VSAT Bande C avec abonnement d'une bande passante de 768 Ko descendante /256 Ko ascendante. A droite, un abri du groupe électrogène à démarrage automatique en cas de coupure du courant REGIDESO.



Figure 8: Antenne VSAT bande C de l'Université du Burundi

En plus de cette antenne VSAT, l'université du Burundi s'est dotée d'une station de base « Broadband », radio access method du type TDD (*Time Division Duplex*), avec une fréquence de 5 Ghz et une bande passante 54 Mbps.



Figure 9 : Figure de la station de base de l'Université du Burundi

Une station de réception située à la Bibliothèque centrale est reliée directement à cette station de base, ce qui permet l'interopérabilité entre les équipements sans-fil de l'ancienne génération et des équipements de la nouvelle génération.

La station de base est montée sur un pylône de 18 m de hauteur avec une protection anti-foudre au sommet.

Sur l'image ci-dessus, on voit au premier plan à gauche la salle polyvalente multimédia et à droite l'antenne VSAT ; derrière l'antenne VSAT, se trouve le bâtiment abritant le serveur. Plus loin entre les deux bâtiments, on voit le pylône de la station de base.

Le réseau fonctionne grâce à la combinaison des équipements de générations différentes : l'antenne omnidirectionnelle posée en 2004 sur le sommet de la toiture de la bibliothèque ainsi que treize stations de réception de la même génération que l'antenne



Figure 9 : Figure de la station de base de l'Université du Burundi

Une station de réception située à la Bibliothèque centrale est reliée directement à cette station de base, ce qui permet l'interopérabilité entre les équipements sans-fil de l'ancienne génération et des équipements de la nouvelle génération.

La station de base est montée sur un pylône de 18 m de hauteur avec une protection anti-foudre au sommet.

Sur l'image ci-dessus, on voit au premier plan à gauche la salle polyvalente multimédia et à droite l'antenne VSAT ; derrière l'antenne VSAT, se trouve le bâtiment abritant le serveur. Plus loin entre les deux bâtiments, on voit le pylône de la station de base.

Le réseau fonctionne grâce à la combinaison des équipements de générations différentes : l'antenne omnidirectionnelle posée en 2004 sur le sommet de la toiture de la bibliothèque ainsi que treize stations de réception de la même génération que l'antenne

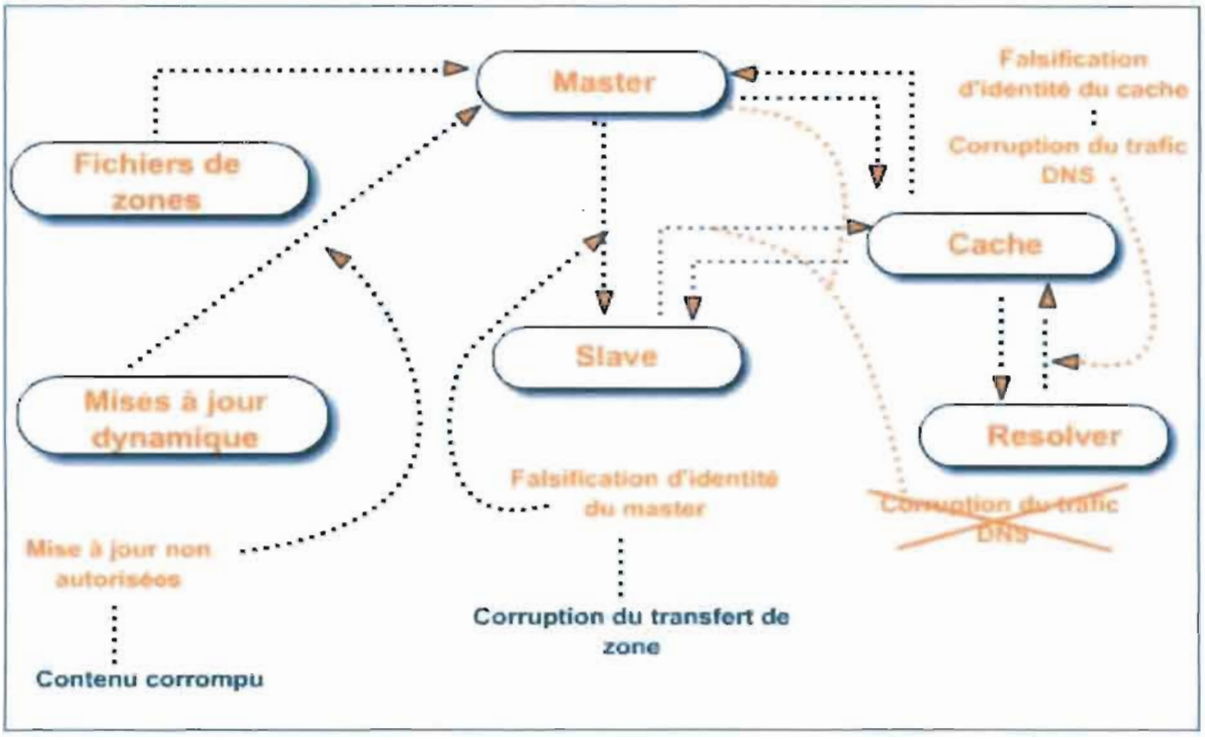


Figure 6 : Schéma de bilan des vulnérabilités résolues par DNSSec

III.2.2. Anticipation de pannes

Un réseau finit toujours par tomber en panne: défaillances matérielles, bogues dans les logiciels, erreurs humaines. Les conséquences peuvent être mineures comme la perte de connexion pour quelques utilisateurs, mais elles sont parfois catastrophiques en guise d'exemple la perte de données importantes ou de travaux fondamentaux.

Puisque le DNS s'appuie fortement sur les réseaux, il est particulièrement vulnérable aux ruptures de connexion. Heureusement, sa conception prend en compte les imperfections du réseau.

Après avoir fait le tour sur l'évolution de la structure du réseau de l'université du Burundi la tâche suivante est de faire une proposition de l'implémentation, de l'exploitation et de la maintenance d'un système de nom de domaine.

La section suivante aborde ces aspects.

IV.3. Implémentation, exploitation et maintenance du serveur de noms de domaine

Le paramétrage de ce serveur de noms de domaine a été fait en utilisant le programme Bind9.

Bind9 a été installé en utilisant la commande **aptitude install bind9.**
Après l'installation de ce programme nous sommes passés à la configuration proprement dite du serveur primaire.
Les étapes suivantes ont été suivies :

1. Nous avons donnés une adresse IP fixe au serveur c'est-à-dire une adresse qui ne change pas;
2. Nous avons crée le fichier de zone **universiteub.bi.hosts** dans le répertoire /etc/bind
3. Ensuite nous avons rempli ce fichier de zone par ces enregistrements :

```
$TTL    604800

@       IN      SOA     ns1.universiteub.bi. admin.universiteub.bi. (

                        1       ; Serial

                        604800      ; Refresh

                        86400       ; Retry

                        2419200     ; Expire

                        604800 )    ; Negative Cache TTL

;

        IN      NS      ns1.universiteub.bi.

@       IN      A       192.168.2.123

ns1     IN      A       192.168.2.123

administrateur-laptop   IN      A       192.168.2.124

mutanga         IN      A       192.168.2.125
```

4. Après la création du fichier de zone nous avons crée le fichier **universiteub.bi.reverse** dans le répertoire /etc/bind pour la résolution inverse .

Ce fichier se présente comme suit :

```
$TTL    604800

@       IN      SOA     ns1.universiteub.bi. admin.universiteub.bi. (

                        1       ; Serial

                   604800       ; Refresh

                    86400       ; Retry

                  2419200       ; Expire

                   604800 )     ; Negative Cache TTL
;
@       IN      NS      ns1.universiteub.bi.

123     IN      PTR     ns1.universiteub.bi.
124     IN      PTR     administrateur-laptop.universiteub.bi.
125     IN      PTR     mutanga.universiteub.bi.
```

Après cela il fallait vérifier la présence de la ligne Include "/etc/bind/named.conf.local" dans /etc/bind/named.conf pour continuer.

Cette ligne étant là nous avons rempli le fichier /etc/bind/named.conf.local par les élements suivants :

```
zone "universiteub.bi" {
        type master;
        file "/etc/bind/universiteub.bi.hosts";
};
zone "2.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/universiteub.bi.reverse";
};
```

En plus de ces éléments mis dans le fichier named.conf.local, nous avons ajouté les élements suivants dans le fichier resolv.conf du répertoire /etc/bind :

```
search universiteub.bi
nameserver 192.168.2.123
```

La configuration du serveur primaire étant finie nous avons alors redémarrer le réseau par la commande /etc/init.d/networking restart ensuite le service bind par la commande /etc/init.d/bind9.
Pour que les modifications opérées sur les fichiers de configurations puissent être prises en compte par bind9, nous avons lancé la commande **rndc relaod.**

A la fin de la configuration nous avons effectué des tests pour vérifier si la configuration s'est bien déroulée. Les commandes suivantes ont été lancées :

host -t a ns1.universiteub.bi

La réponse a été:
ns1.universiteub.bi has address 192.168.2.123

Ceci nous a montré que la tâche de configuration du serveur primaire a été fait avec succès. Le serveur a pour nom ns1, se trouve dans le domaine universiteub.bi et a pour adresse 192.168.2.123.

Après avoir constaté que le serveur tourne, nous avons paramétré un client c'est-à-dire une machine du réseau qui n'est pas serveur pour vérifier que son nom sera résolu en adresse IP.
Sa configuration s'est faite en ajoutant dans le fichier /etc/resolv.conf les lignes suivantes :

search universiteub.bi
nameserver 192.168.2.123

Le mot réservé search indique au client le domaine auquel il fait partie et nameserver lui indique l'adresse IP du serveur.
Comme nous l'avons fait pour le serveur, nous avons donné à cette machine cliente l'adresse IP fixe 192.168.2.124.

Après avoir configuré le client il a fallu redémarrer le réseau.

Sur ce client des tests de bon fonctionnement ont été menés avec succès. Il s'agit d'un ping qui vérifie que les paquets envoyés sur le serveur l'atteignent ou sur toute autre machine du réseau; du test nslookup qui permet de vérifier si le système parvient à résoudre l'adresse IP du client en son nom ou inversement.

Après la configuration du serveur principal nous avons paramétré le serveur secondaire qui prendra le relais en cas de pépin du serveur primaire.

Les lignes suivantes ont été ajoutées dans le fichier named.conf.local du serveur secondaire :

```
zone "universiteub.bi" {
    type slave;
master {192.168.2.123;};
    file "/etc/bind/universiteub.bi.hosts";
};
```

Du point de vue exploitation du DNS, étant donné que nous ne sommes pas responsables de la gestion du réseau, nous nous sommes limité à l'ajout des clients sur notre serveur et nous avons procédé à des tests de bon fonctionnement La tâche essentielle de gestion de ce DNS reviendra à l'administrateur réseau qui est appelé à suivre l'évolution quotidienne du réseau.

C'est donc un travail permanant. Néanmoins grâce au programme de gestion de syllabus au moyen d'un serveur web que le consultant Ephrem SEBATIGITA était en train de mettre en place, nous avons pu mettre en application le DNS.

En effet pour joindre les syllabus sur ce serveur il n'est plus nécessaire de les appeler par adresse IP du serveur. Ils sont accessibles sur le nom du domaine.

Quant à la maintenance du dit DNS, nous avons prévu un serveur secondaire qui prendra la relève en cas de panne du serveur primaire ce qui garantit la permanence du service DNS au sein du réseau.

Avant de conclure ce travail, il est essentiel de faire le tour des différents outils d'administration des réseaux informatiques qui nous ont permis de vérifier que notre objectif a été atteint. La section suivante en fait le point.

IV.4. Les principaux outils pour tester le bon fonctionnement d'un réseau

IV.4.1. L'outil ping

C'est un outil qui permet de vérifier si les paquets envoyés par un ordinateur du réseau atteignent un autre ordinateur du même réseau ou d'un réseau autre que celui comprenant l'expéditeur des paquets avec la condition que les deux soient interconnectés. Ce test permet également de savoir le temps mis par les paquets pour atteindre la machine destinataire ainsi que le nombre de paquets reçus et perdus en cours de route.
Elle s'exécute en mode commande.

Sa syntaxe est la suivante :

Ping <nom_de_la_machine> ou ping <adresse>

Ces deux commandes donnent le même résultat et dans le cadre de notre travail, nous avons trouvé un résultat satisfaisant à partir du nom ce qui nous montre que notre service DNS a été bien implémenté. Les résultats de ce test sont consigné dans l'annexe 1.

IV.4.2. L'outil nslookup

C'est un outil qui permet de vérifier si le serveur DNS parvient à résoudre le nom d'une machine du réseau en son adresse IP et inversement.
Elle s'exécute en mode commande et sur n'importe quelle machine du réseau.
Sa syntaxe est la suivante : **nslookup <nom_de_la_machine>**
Cette commande renvoie l'adresse du serveur DNS, le nom FQDN de la machine et son adresse IP.
Grâce à cette commande, notre serveur DNS a pu résoudre le nom d'une machine du réseau. Les résultats de ce test sont donnés dans l'annexe 2.

IV.4.3. L'outil Traceroute

C'est un outil qui permet de retracer la route suivie par les paquets depuis l'expéditeur jusqu'au destinataire. Si les paquets ne parviennent pas à atteindre le destinataire, nous parvenons à identifier à quel endroit les paquets sont arrivés et ainsi savoir à quel niveau intervenir. Sa syntaxe est la suivante : **Traceroute <nom_FQDN>**
Le résultat de ce test sont repris dans l'annexe 3.

IV.4.4. Les outils ifconfig et netstat

La commande ifconfig permet de connaître la configuration IP, l'usage du DHCP, l'adresse IP, le masque de sous réseau pour chaque carte réseau.
Sa syntaxe est la suivante : **ifconfig**
Les résultats de cette commande sont repris à l'annexe 4 de ce travail.

Quant à la commande netstat, elle permet d'afficher, avec l'argument –nr les tables de routage qui indiquent les routes disponibles pour les paquets depuis le réseau. Avec l'argument –i, elle permet d'afficher les statistiques réseaux qui sont importantes pour pouvoir analyser la charge du réseau.
Leurs syntaxes sont respectivement **netstat –nr** et **netstat –i.**
Les résultats de ces tests sont repris aux annexes 4 et 5.

En plus de ces outils de test d'un réseau il existe d'autres outils qui permettent l'administration et l'exploitation d'un réseau. Les plus connus et couramment utlisés sont FTP et TELNET.

IV.4.5. Les utilitaires FTP et TELNET

FTP (File Transfert Protocol) est un protocole, c'est-à-dire un langage standard de communication entre deux machines, permettant à des machines de types différents de transférer des fichiers sur un réseau fonctionnant sous TCP/IP.

Le protocole FTP permet d'échanger un fichier à la fois, dans les deux sens entre la machine client (celle qui a initié la connexion, donc la machine appelante) et la machine serveur (celle qui fournit le service FTP, donc la machine appelée). Le protocole FTP permet aussi d'autres actions telles que la création et la suppression et le renommage de fichiers, etc.

La commande pour initier une session FTP est la suivante : ftp nom_du_serveur
Pour envoyer un fichier, la commande utilisée est : put nom_du_fichier.

La commande TELNET quant à lui permet de se connecter sur une machine distante. Le login (compte utilisateur) et le mot de passe sont demandés lors de la connexion.

La commande TELNET est utilisé comme suit : telnet nom_de_la_machine_distante

CONCLUSION GENERALE

A l'issue de notre travail qui a pour titre ***«Etude de la gestion d'un système de noms de domaine (DNS) à l'université du Burundi : implémentation, exploitation et maintenance»***, il importe de passer en revue les principaux aspects qui ont caractérisé ce travail.

L'objectif poursuivit était l'implémentation du serveur DNS au sein du réseau informatique de l'université du Burundi pour permettre la localisation des machines non par leurs adresses IP difficile à retenir mais par des noms ayant un sens significatif facile à retenir pour les utilisateurs du réseau.

Le chapitre premier a passé en revu l'historique et les principes du DNS.

Sur un réseau les machines communiquent entre elles normalement par des adresses IP représentés en format décimal pointé en utilisant un protocole appelé TCP/IP.

Cette notation numérique pour identifier les machines n'est pas commode pour l'utilisateur qui a du mal à retenir une longue suite de chiffres. Il est plus aisé de leur attribuer des noms qui vont servir à les identifier.

Ainsi est né un service de mise en correspondance nom - adresse numérique (adresse IP) qui permet aussi bien de trouver l'adresse IP d'une machine à partir de son nom (résolution de nom) que de retrouver le nom à partir de l'adresse IP (résolution inverse).

Ce service de mise en correspondance était assuré au départ à travers un fichier unique centralisé nommé **hosts.txt** dans lequel étaient mentionnés les noms de toutes les machines connectées au réseau ainsi que leurs adresses IP correspondantes.

Ce fichier devait être téléchargé par FTP et stocké dans un fichier standard (**/etc/hosts**) consulté pour effectuer la résolution de nom et la résolution inverse.

Ce procédé, qui était relativement efficace dans les années 70 a vite atteint ses limites avec l'expansion du réseau auquel sont aujourd'hui connectées plus de cent millions de machines.

Au cours du second chapitre, nous avons fait le contour de tous les aspects relatifs à l'implémentation du DNS. Ce chapitre explique les différents enregistrements du fichier de zone et la syntaxe de chacun d'eux.

En plus nous avons montré comment se fait le démarrage des serveurs primaire et secondaire.

Le troisième chapitre relate les aspects ayant trait à l'exploitation et la maintenance du DNS. Ces aspects sont entre autres la gestion des différents sous domaines, la délégation d'autorité, l'anticipation de pannes éventuelles, l'expansion du système et le traitement des pannes le cas échéant.



Ce fichier devait être téléchargé par FTP et stocké dans un fichier standard (**/etc/hosts**) consulté pour effectuer la résolution de nom et la résolution inverse.

Ce procédé, qui était relativement efficace dans les années 70 a vite atteint ses limites avec l'expansion du réseau auquel sont aujourd'hui connectées plus de cent millions de machines.

Au cours du second chapitre, nous avons fait le contour de tous les aspects relatifs à l'implémentation du DNS. Ce chapitre explique les différents enregistrements du fichier de zone et la syntaxe de chacun d'eux.

En plus nous avons montré comment se fait le démarrage des serveurs primaire et secondaire.

Le troisième chapitre relate les aspects ayant trait à l'exploitation et la maintenance du DNS. Ces aspects sont entre autres la gestion des différents sous domaines, la délégation d'autorité, l'anticipation de pannes éventuelles, l'expansion du système et le traitement des pannes le cas échéant.

le réseau et de l'autre l'absence de schéma de l'architecture du réseau ne le permettait pas.

Tout au long de notre travail, nous avons constaté que le réseau n'est exploité que pour le partage de l'internet alors qu'il devrait servir au partage de ressources et de données.

Au vu de ce rôle, nous recommandons à l'Université du Burundi de mettre en place le serveur de fichiers et de s'investir dans la mise en place d'une culture de travail en réseau et d'archivage électronique.

Enfin, bien que nous soyons satisfait du travail accompli qui nous a permis d'apprendre et de comprendre le fonctionnement du DNS, nous ne prétendons pas avoir épuisé tous les aspects du sujet. C'est pourquoi si l'opportunité nous est offerte nous avons l'ambition de le poursuivre et nous invitons aussi aux futurs chercheurs de nous compléter.

REFERENCES BIBLIOGRAPHIQUES

I. OUVRAGES GÉNÉRAUX

1. APREA J.F, *DNS : concepts, architecture et administration sous windows,* Editions ENI, 2005.
2. ALBITZ P. & CRICKET L., *DNS et BIND*, Editions O'REILLY, 2006.
3. SCRIMGER R. & KELLY A., *Préparation aux MCSE.*
4. JACK T., Jr & STEVEN B., *Linux, installation, configuration et administration des systèmes linux,* Campus press.
5. CRAIG H. (Traduit par ERIC DUMAS), *TCP/IP, Administration de réseau*, 2ème éd., Editions O'REILLY, Paris 1998.
6. LOTTOR M (1981-1991), *RFC, Internet Growth*, SRI International

II. DOCUMENTS ELECTRONIQUES ET AUTRES

1. **http://www.frameip.com/dns/**
2. **http://doc.ubuntu-fr.org/**
3. **http://www.afnic.fr/doc/formations/autormation/dns/**
4. **http://www.theodys.org/index.php/**
5. **Université du Burundi, *Schéma d'informatisation de l'Université du Burundi.***

ANNEXES

Annexe 1 : Résultats du test ping sur l'hôte PC1SQC

melanie@melanie-laptop:~$ ping PC1SQC

PING PC1SQC.universiteub.bi (192.168.1.22) 56(84) bytes of data.
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=1 ttl=128 time=0.321 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=2 ttl=128 time=0.334 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=3 ttl=128 time=0.336 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=4 ttl=128 time=0.340 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=5 ttl=128 time=0.335 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=6 ttl=128 time=0.322 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=7 ttl=128 time=0.322 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=8 ttl=128 time=0.338 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=9 ttl=128 time=0.333 ms
64 bytes from PC1SQC.universiteub.bi (192.168.1.22): icmp_seq=10 ttl=128 time=0.344 ms

--- PC1SQC.universiteub.bi ping statistics ---

10 packets transmitted, 10 received, 0% packet loss, time 8997ms

rtt min/avg/max/mdev = 0.321/0.332/0.344/0.019 ms

Annexe 2: Résultat du test nslokup sur l'hôte PC1SQC

melanie@melanie-laptop:~$ nslookup PC1SQC

Server: 192.168.1.106
Address: 192.168.1.106#53

Name: PC1SQC.universiteub.bi
Address: 192.168.1.22

Annexe 3: Résultat du test traceroute sur l'hôte PC1SQC

melanie@melanie-laptop:~$ traceroute PC1SQC

traceroute to PC1SQC (192.168.1.22), 30 hops max, 40 byte packets
1 PC1SQC.universiteub.bi (192.168.1.22) 1.501 ms 1.516 ms 1.573 ms

Annexe 4: Résultat du test ifconfig sur le serveur DNS

melanie@melanie-laptop:~$ ifconfig

eth0 Link encap:Ethernet HWaddr 00:1b:38:4b:9c:b5

inet adr:192.168.1.106 Bcast:192.168.1.255 Masque:255.255.255.0

adr inet6: fe80::21b:38ff:fe4b:9cb5/64 Scope:Lien

UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1

Packets reçus:2582 erreurs:0 :0 overruns:0 frame:0

TX packets:4033 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 lg file transmission:1000

Octets reçus:646264 (631.1 KB) Octets transmis:585877 (572.1 KB)

Interruption:217 Adresse de base:0xe000

lo Link encap:Boucle locale

inet adr:127.0.0.1 Masque:255.0.0.0

adr inet6: ::1/128 Scope:Hôte

UP LOOPBACK RUNNING MTU:16436 Metric:1

Packets reçus:2707 erreurs:0 :0 overruns:0 frame:0

TX packets:2707 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 lg file transmission:0

Octets reçus:146523 (143.0 KB) Octets transmis:146523 (143.0 KB)