Table des matières

Liste des illustrations

GNU/Linux Services Serveurs

1. Avertissement

Ce document accompagne la distribution "freeduc-sup" que vous pouvez télécharger *sur le site officiel : ici* (http://freeduc-sup.eu.org).

La distribution et les supports qui l'accompagnent ont été conçus pour nos besoins. Ils sont également à votre disposition.

En les utilisant vous n'engagez que vous.

Tout n'est pas encore finalisé à ce jour : 08/09/2003, date de dernière modification.

Vous pouvez m'écrire directement pour les remarques. mascret@linux-france.org.

Ont participé pour la réalisation de la distribution freeduc-sup et de ce document : Valérie Émin, Ludovic Grossard, Jean Philippe Gaulier, Sylvain Cherrier, Olivier Capuozzo, Remi Bernhard.

2. Date de dernière modifi cation

2003-09-22

3. En cours de réalisation

LDAP, le TP authentification LDAP est fait mais non relu. C'est pour bientôt.

4. Les archives

Il y a des documents à fournir aux étudiants pour les TP sur HTTP (cgi) et PostgreSQL. Vous les aurez sous les noms respectifs de "docTP_Apache.tar" dans le répertoire correspondant au TP et l'autre sous le nom de "tp_postgresql.tar" dans le répertoire correspondant au TP PostgreSQL.

5. Résumé

Ce document décrit une procédure qui va permettre de procéder à l'installation de l'ensemble des services serveurs sur un intranet. La mise en ouvre est réalisée avec des logiciels libres. Les différents TP permettent

l’apprentissage des savoirs (S1-S2) décrits dans le référentiel du BTS informatique de Gestion option Administrateur de réseau local.

Éléments de cours sur TCP/IP

Résumé sur TCP/IP

1. Présentation

Ce document présente quelques rappels sur le protocole TCP/IP. Il est essentiel de bien maîtriser ces aspects généraux, l'installation et la configuration du protocole TCP/IP, avant d'aborder les TP(s) sur la mise en oeuvre des services réseaux comme le routage, NFS, les Rcommandes... Il décrit le modèle en couches du DOD, les classes d'adresses, le fonctionnement du protocole ARP, le rôle des ports et des sockets.

2. Historique du protocole TCP/IP

Ce protocole de communication a été mis au point à partir d'une étude commandée au début des années 1970 par le DARPA (Defense Advanced Project Research Agency) dépendant du DoD (Department of Defense) Américain. L'objectif était de mettre au point un protocole de communication permettant d'interconnecter les ordinateurs de toutes marques dont disposait l'armée des US.

Premières implémentations au début des années 1980. Elles introduisaient les notions de couches de communication. Le protocole TCP/IP n'est pas normalisé OSI, même si aujourd'hui il est largement plus utilisé que le protocole OSI.

2.1. Principales raisons du succès de TCP/IP

Il est intégré dans les systèmes Unix ce qui en a assuré une grande diffusion. Les spécifications sont du domaine public, et elles sont facilement accessibles sur des serveurs de fichiers internationaux, ce qui a permis de nombreux développements dans les milieux universitaires et de la recherche. Les spécifications sont fournies sous la forme de RFC (Request for Comments).

Il est maintenant disponible sur la plupart des plates-formes matérielles et systèmes d'exploitation (il est même de plus en plus souvent livré avec le système) de l'ordinateur personnel (PC ou Mac) au plus gros calculateur vectoriel (Cray,...

Il est utilisable sur la plupart des réseaux physiques (Ethernet 802.3 , Token Ring 802.5, liaisons séries) et même à travers d'autres réseaux publics (X25, Numéris).

De très nombreux logiciels ont été développés sur TCP/IP, qu'ils soient du domaine public ou vendus par des sociétés spécialisées.

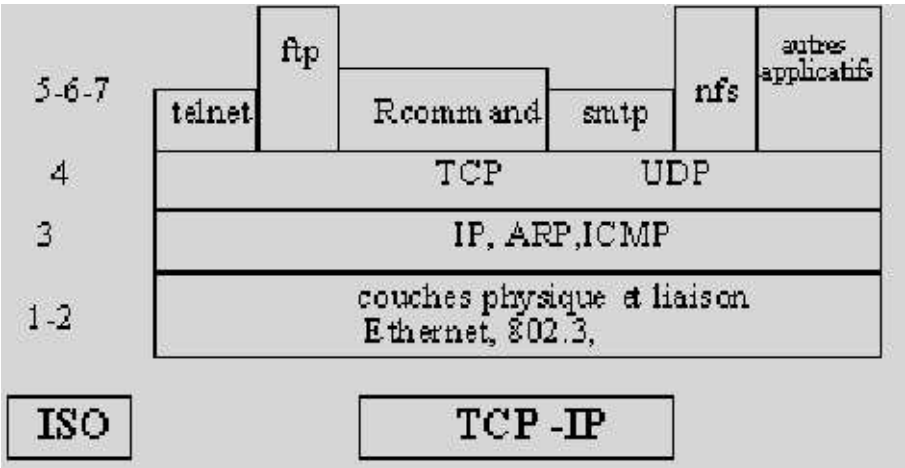
3. Les couches IP et TCP

Ce modèle en 4 couches (Application, Transport, Réseau, Physique) est parfois appelé modèle DOD.

3.1. Les principaux composants de la pile TCP/IP sont les suivants

- IP (Internet Protocol) : C'est un protocole de niveau 3. Il assure le transfert des paquets TCP/IP sur le réseau local, et avec les réseaux extérieurs via des routeurs. Le protocole IP travaille en mode non connecté, c'est-à-dire que les paquets émis par le niveau 3 sont acheminés de manière autonome (datagrammes), sans garantie de livraison.
- ARP (Address Resolution Protocol) : Protocole qui permet d'associer l'adresse de niveau 3 (ie @ip) à une adresse de niveau 2 (par ex Ethernet)
- ICMP (Internet Control and error Message Protocol) : Utilisé pour les tests et les diagnostics
- TCP (Transport Control Protocol) : Protocole de niveau 4 qui fonctionne en mode connecté. Sur une connexion TCP entre deux machines du réseau, les messages (paquets ou segments TCP) sont acquittés et délivrés en séquence.
- UDP (User Datagram Protocol) : Protocole de niveau 4 en mode non connecté : les messages (ou paquets UDP) sont transmis de manière autonome.

Figure 1. Pile de protocole IP



3.2. Quelques applications utilisées en environnement TCP/IP

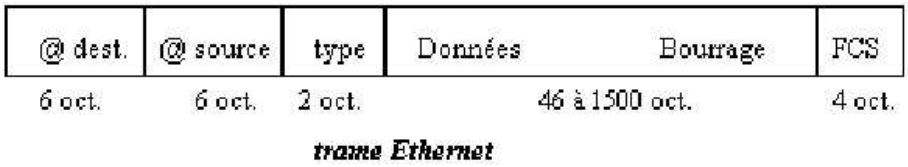
- r-commands : (ou remote commandes) : exécution d'une commande à distance sur une autre machine du réseau local
- telnet : connexion interactive
- ftp (File Transfert Protocol) : transfert de fichiers
- smtp (Simple Mail Tranfert Protocol) : messagerie
- nfs : (Network File System): système de fichiers répartis

Sur un même réseau physique (Ethernet par exemple) le protocole TCP/IP peut cohabiter avec d'autres protocoles de niveau 3. Pour cela dans la trame de niveau 2 un champ identifie le type de protocole de niveau 3.

3.3. Protocole, pilote, interface

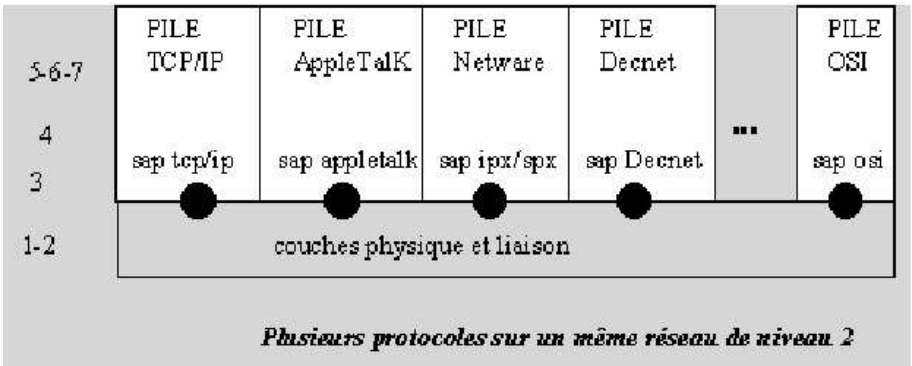
Plusieurs protocoles peuvent même cohabiter sur une même machine : le niveau 2 est géré par le pilote (driver) de la carte (Ethernet par ex), au dessus duquel il y a plusieurs "piles" de niveau supérieur. Le paquet extrait de la trame est transmis à la pile correspondant au type de protocole (cf notion de SAP - Service Access Point - du modèle OSI)

Figure 2. Schéma d'une trame Ethernet



trame Ethernet

Figure 3. Les piles de protocoles



Plusieurs protocoles sur un même réseau de niveau 2

4. Les adresses TCP/IP

L'adresse TCP/IP d'une machine est une adresse de niveau réseau codée sur 32 bits (ie 4 octets en IPv4) qui est en général notée sous la forme de 4 chiffres séparés par des points. On parle de notation en décimale pointé. Chaque champ, qui représente un octet, peut prendre des valeurs entre 0 et 255.

Exemple : *192.93.116.3*

4.1. Structure d'une adresse IP

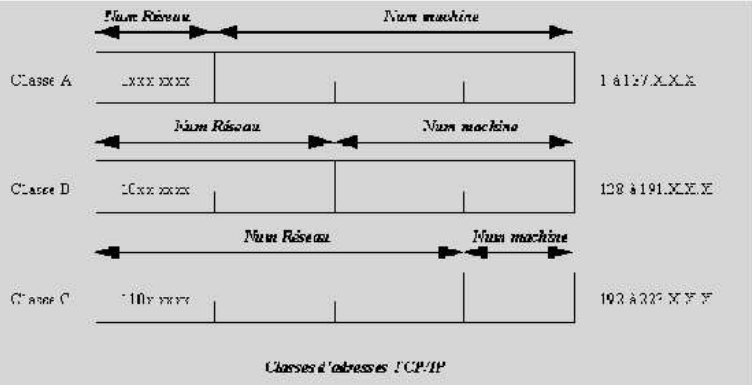
L'adresse IP est constituée d'un champ *numéro de réseau* (1, 2 ou 3 octets) et d'un champ *numéro de machine dans le réseau* (3, 2 ou 1 octets). L'adresse ip = adresse de réseau + adresse de machine.

L'adresse de réseau est attribuée par un organisme officiel : le NIC aux US (ou ses représentants)

L'adresse de machine est attribué localement par le gestionnaire du réseau (nota: il est possible de découper le champ de droite en *sous-réseau+machine*. Les réseaux TCP/IP sont rangés en 3 classes A, B ou C en fonction de la taille du champ numéro de réseau:

- classe A : 1 à 127.X.X.X
- classe B : 128 à 191.X.X.X
- classe C : 192 à 223.X.X.X (les adresses > à 223 sont réservées à d'autres usages)

Figure 4. Les classes d'adresses



4.2. Utilisation des adresses IP

Le nombre de machines dans le réseau dépend donc de la classe du réseau. Chaque octet du champ machine peut prendre des valeurs entre 1 et 254. Les valeurs 0 (tous les bits à 0) et 255 (tous les bits à 1) sont réservées :

Un champ machine tout à 0 sert à désigner le numéro de réseau (notamment pour le routage)

Un champ tout à 1 indique un message de broadcast adressé à toutes les machines IP du réseau.

Sur les fichiers de configuration on a un masque réseau (netmask) qui, associé à l'adresse IP, indique le champ à prendre en compte pour le *réseau* (bits à 1), et celui à prendre en compte pour la *machine* (bits à 0).

Exemple : dans un réseau de classe A sans sous-réseau : netmask=255.0.0.0

4.3. Les adresses réservées

- 0.0.0.0 est réservée pour la route par défaut. L'adresse désigne tous les réseaux. Tous les paquets destinés à un réseau inconnu, seront dirigés vers cette route.
- 127.0.0.0 est réservée au trafic IP de la machine locale. Une interface locale porte en général l'adresse 127.0.0.1 appelée adresse de "loopback"
- Certaines adresses peuvent également être librement utilisées pour monter un réseau privé. Voici les adresses : Classe A : 10.0.0.0, Classe B : 172.16.0.0 à 172.31.0.0, Classe C : 192.168.0.0 à 192.168.255.0

Aucun paquet provenant de ces réseaux ou à destination de ces réseaux, ne sera routé sur l'Internet.

4.4. Types d'utilisation des adresses IP

Il y a 3 modes d'utilisation des adresses :

- L'adressage d'un noeud ou d'un hôte directement. On parlera d'unicast.
- L'adressage de n noeuds simultanément. On parlera de multicast. Cette technique est utilisée pour des applications de visio-conférence par exemple. On utilise des adresses supérieures à 223.x.y.z (224.0.0.9 par exemple) appelées parfois adresses de classe D.
- L'adressage de tous les noeuds d'un réseau, on parlera de broadcast.

5. Sous-réseaux et adresses IP sans classe

Ce paragraphe est inspiré des informations données dans l'ouvrage "TCP/IP, Administration de réseau" et édité chez O'Reilly.

5.1. La notation CIDR

Plutôt que de conserver une identification des réseaux orientée "octets" à la façon de la notation en décimale pointé, et en conservant les classes d'adresses A, B, C, la notation CIDR (Classless Inter Domain Routing) propose une identification réseau/hôte orientée "bit". Avec ces masques de bits, il n'y a plus la limitation liée aux classes.

Cela à une incidence sur le routage et sur les tables de routage. Prenons par exemple 256 réseaux de classe C d'adresses contigües (192.168.0.0 à 192.168.255.0). Il faudrait 256 routes sur un routeur pour identifier tous ces réseaux de classe C. Avec l'identification binaire, une seule route suffit. Il suffit d'adresser un réseau 192.168.0.0/255.255.0.0.

Cette technique s'appelle du "supernetting" ou "masquage de sur-réseaux" et allège considérablement les tables de routage et leur maintenance..

C'est pour cette raison que des plages d'adresses groupées comme 194.0.0.0 à 195.255.255.ont été affectées à l'Europe.

Cela à également une incidence sur le fonctionnement (routage) des équipements réseaux et sur les protocoles. Ceux-ci doivent prendre en charge le masquage binaire.

Plutôt que de noter une adresse sous la forme adresse/masque (par exemple 192.168.0.1/255.255.255.0), la notation CIDR propose une forme adresse/préfixe (par exemple 192.168.0.1/24), qui signifie que l'on masque ici sur 24 bits.

5.2. Les sous-réseaux

Le masquage de sous-réseaux, ou subnetting, consiste à utiliser la partie affectée normalement à l'hôte, pour segmenter la plage d'adresses disponibles en sous-réseaux.

Prenons l'exemple d'une personne qui souhaiterait créer des sous réseaux à partir d'une classe A 10.0.0.0/8. La partie hôte est représentée par les 3 derniers digits, soit les 24 bits de droite. En masquant sur 8 bits, la personne s'autorise 2 ^ 8 sous réseaux. Les adresses deviendront 10.x.y.z/16, où x représente le sous-réseau dans le réseau 10, et y.z, l'adresse de l'hôte dans le sous-réseau x.

En théorie il est possible de choisir les bits que l'on veut. En pratique on utilise les bits les plus à gauche de la section masquable.

Le fonctionnement des sous-réseaux est décrit dans la RFC 1878. Il était alors traditionnellement conseillé de ne pas utiliser les sous-réseaux ayant tous les bits à 1 ou tous les bits à 0.

Cela génère une perte importante d'adresse. Un masque de /26 (255.255.255.192) sur une classe C fait perdre 2 adresses de sous-réseaux. Le x.y.z.0/26 et le x.y.z.192/26.

Ces sous-réseaux sont, selon la RFC 1812 de 1995 complètements valides et doivent être pris en charge par les équipements de routage.

Ceci est le cas des nouveaux équipements de routage. On considèrera donc que c'est la RFC 1812 qui servira de référence.

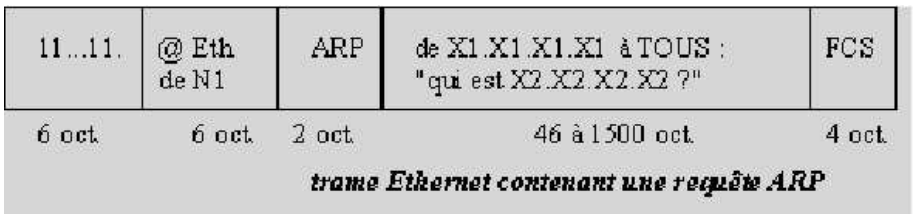
6. Le protocole ARP

L'adresse Ethernet est une adresse unique sur 48 bits (6 octets) associée à la carte Ethernet. Lorsqu'un noeud N1 du réseau TCP/IP X1.X1.X1.X1 veut émettre un paquet TCP/IP (dans une trame Ethernet) vers une machine N2 d'adresse IP (X2.X2.X2.X2), il faut qu'il connaisse l'adresse Ethernet (E2.E2.E2.E2.E2.E2). Pour réaliser l'association *@ip / @ Ethernet* l'émetteur N1 utilise le protocole ARP dont le principe est le suivant :

L'émetteur envoie une trame Ethernet de diffusion (broadcast) (ie @destinataire toute à 1) contenant un message ARP demandant

qui est X2.X2.X2.X2 ?

Figure 5. Trame Ethernet contenant une requête ARP



Toutes les machines IP du réseau local reçoivent la requête. N2 qui a l'adresse X2.X2.X2.X2 se reconnaît, et elle répond à N1 ie X1.X1.X1.X1 (dans une trame destinée à E1.E1.E1.E1.E1.E1)

Figure 6. Trame Ethernet contenant une réponse ARP

Chaque machine maintient en mémoire une table cachée de correspondances *@ip / @ Ethernet* pour éviter trop de requêtes ARP. Chaque entrée de la table à une durée de vie limitée. Voici pour exemple ce que donne le programme tcpdump avec la commande "ping 192.168.1.2" à partir de la machine uranus alors que la table arp de l'hôte uranus est vide:

- 13:17:14.490500 arp who-has 192.168.1.2 tell uranus.planete.net
- 13:17:14.490500 arp reply 192.168.1.2 is-at 0:40:33:2d:b5:dd
- 13:17:14.490500 uranus.planete.net > 192.168.1.2: icmp: echo request
- 13:17:14.490500 192.168.1.2 > uranus.planete.net: icmp: echo reply
- 13:17:15.500500 uranus.planete.net > 192.168.1.2: icmp: echo request
- 13:17:15.500500 192.168.1.2 > uranus.planete.net: icmp: echo reply

Explications :

Ligne 1 uranus demande qui est 192.168.1.2 (requête ARP) Le paquet est diffusé à tous les hôtes du réseau.

Ligne 2 réponse ARP: je suis à l'adresse Ethernet 00:40:33:2d:b5:dd

Lignes 3 à 6 : Echanges de paquets ICMP entre les 2 hôtes.

6.1. Les domaines et les noms de machine

Il est peu commode de désigner une machine par son adresse IP. On peut aussi utiliser un nom qui se présente en général sous la forme *nom_machine* (ex uranus) ou *nom_machine.sous_domaine.domaine* (ex : uranus.toubet.edu). C'est quand même l'adresse IP qui est utilisée en interne dans les paquets au cours des échanges. Pour cela il faut un mécanisme qui permette de traduire le nom_machine en adresse IP.

Deux solutions sont utilisées :

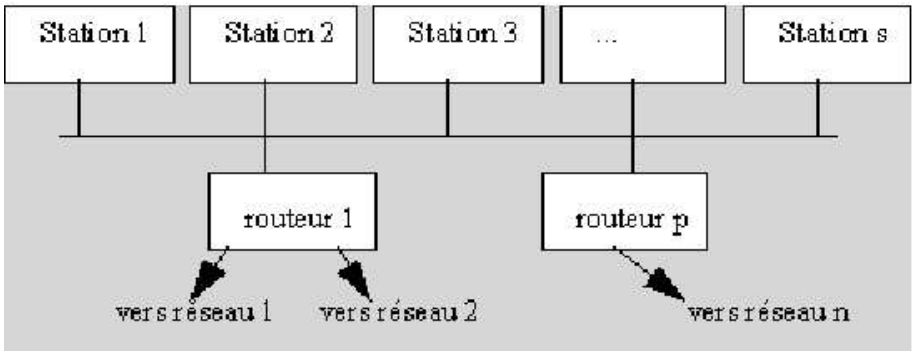
- Noms locaux : Sur chaque machine on crée un fichier qui contient la table de correspondance nom_machine --- @ip (par ex le fichier /etc/hosts sur un système Unix)
- Serveurs de noms : pour chaque domaine (par ex toubet.edu) une machine serveur de noms (serveur DNS) contient l'annuaire des machines du domaine. Les machines des utilisateurs sont configurées pour interroger le serveur. Il y a en général plusieurs serveurs DNS pour un même domaine au cas ou le serveur primaire tomberait en panne.

Il est également possible de combiner les deux solutions.

6.2. Les passerelles ou routeurs

Les réseaux IP sont interconnectés par des routeurs IP de niveau 3 (appelés abusivement en terminologie IP des gateways ou passerelles). Chaque station IP doit connaître le routeur par lequel il faut sortir pour pouvoir atteindre un réseau extérieur, c'est-à-dire avoir en mémoire une table des réseaux et des routeurs.

Figure 7. Réseau et routeur



Commentaires :

- Réseau 1 --> Routeur 1
- Réseau 2 --> Routeur 1
- .....
- Réseau n --> Routeur p

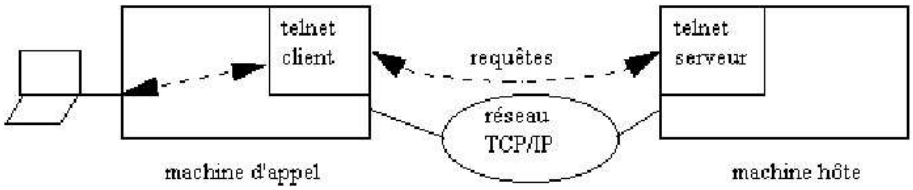
Les tables de routage peuvent être statiques dans le cas de réseaux simples, ou dynamiques dans le cas de réseaux maillés. Le protocole d'échange dynamique des tables IP sur un réseau local est *RIP* (Routing Information Protocol) ou le protocole OSPF.

7. Quelques applications

7.1. Le modèle client/serveur

Les applications réseaux fonctionnent sur le modèle client/serveur. Sur la machine serveur un processus serveur (daemon) traite les requêtes des clients. Client et Serveur dialoguent en échangeant des messages qui contiennent des requêtes et des réponses. Par exemple telnet.

Figure 8. traitement client/serveur

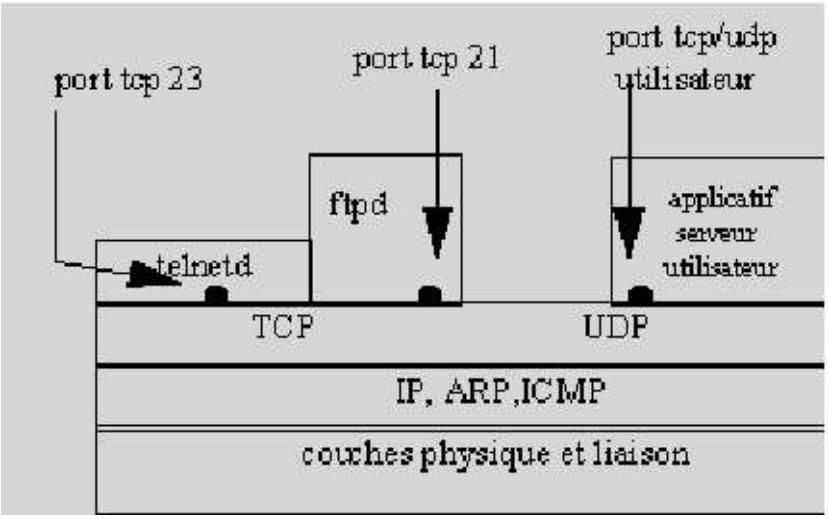


7.2. Adressage des applicatifs

Sur la machine cliente, l'utilisateur (usager ou programme) effectue une requête vers une machine IP serveur sur le réseau. (par exemple "telnet host" ou "ftp host"). Cela se traduit par la réservation d'un port de sortie TCP ou UDP et l'envoi d'un paquet ip à la machine serveur. Ce paquet contient un message TCP ou UDP avec un numéro de port correspondant à l'application demandée sur le serveur.

Sur le serveur, la requête est réceptionnée par le pilote IP, aiguillée vers TCP ou UDP puis vers le port demandé. Le processus serveur correspondant est à l'écoute des appels sur ce port (par exemple le daemon "telnetd" traite les requêtes "telnet", le daemon "ftpd" traite les requêtes "ftp"). Processus client et processus serveur échangent ensuite des messages. des numéros de port sont réservés pour les applications "standards", d'autres sont disponibles pour les applications développées par les utilisateurs. Vous pouvez consulter les ports standards, utilisés par les applications, dans le fichier "/etc/services".

Figure 9. Application et port de communication



Une fois la connexion établie entre le client et le serveur, ceux-ci peuvent s'échanger des informations selon un protocole défini selon l'applicatif. Le client soumet des requêtes auxquelles répondra le serveur.

Ce mode de communication s'appuie sur la couche "socket". Cette couche est une interface entre la couche présentation et transport. Elle permet la mise en place du canal de communication entre le client et le serveur.

On peut schématiquement dire qu'un socket fourni un ensemble de fonctions. Ces fonctions permettent à une application client/serveur d'établir un canal de communication entre 2 ou plusieurs machines, qui utilisent un protocole de transport (TCP ou UDP) et un port de communication.

Fichiers de configuration et commandes de base

Présentation des principaux fichiers de configuration et des commandes d'administration système et réseau.

1.

Les outils de l'administrateur réseau

Présentation du document:

Ce document présente les principaux fichiers de configuration d'une machine en réseau, et les commandes d'administration réseau.

Il est composé de 6 parties:

1. Les fichiers de configuration
2. La commande ifconfig
3. La commande arp
4. La commande route
5. La commande netstat
6. La commande traceroute

2. Les fichiers de configuration

2.1. Le fichier /etc/hosts

Le fichier hosts donne un moyen d'assurer la résolution de noms

Exemple de fichier host

```
127.0.0.1  localhost localhost.localdomain
192.168.1.1  uranus.foo.org uranus
```

2.2. Le fichier /etc/networks

Il permet d'affecter un nom logique à un réseau

```
localnet   127.0.0.0
foo-net 192.168.1.0
```

Cette option permet par exemple d'adresser un réseau sur son nom, plutôt que sur son adresse.

route add foo-net au lieu de route add -net 192.168.1.0

2.3. Le fichier /etc/host.conf

Il donne l'ordre dans lequel le processus de résolution de noms est effectué. Voici un exemple de ce que l'on peut trouver dans ce fichier:

order hosts,bind

La résolution est effectuée d'abord avec le fichier host, en cas d'échec avec le DNS.

2.4. Le fichier /etc/resolv.conf

Il permet d'affecter les serveurs de noms.

Exemple

```
Nameserver 192.168.1.1
Nameserver 192.168.1.2
Nameserver 192.168.1.3
```

Ici le fichier déclare le nom de domaine et 3 machines chargées de la résolution de noms.

2.5. Les fichiers de configuration des interfaces réseau

Vous trouverez ces fichiers dans /etc/network/interfaces. Voici un exemple qui contient 3 interfaces.

```
# /etc/network/interfaces -- configuration file for ifup(8), ifdown(8)
```

```
# The loopback interface
# automatically added when upgrading
auto lo eth0 eth1

iface lo inet loopback


iface eth0 inet static
        address 192.168.90.1
        netmask 255.255.255.0
        network 192.168.90.0
        broadcast 192.168.90.255
        gateway 192.168.90.1

iface eth1 inet static
        address 192.168.0.1
        netmask 255.255.255.0
        network 192.168.0.0
        broadcast 192.168.0.255
```

3.

3.1. La commande ifconfi g

La commande ifconfig permet la configuration locale ou à distance des interfaces réseau de tous types d'équipements (unité centrale, switch, routeur). La ligne de commande est:

ifconfig interface adresse [parametres]

Exemple: ifconfig eth0 192.168.1.2 (affecte l'adresse 192.168.1.2 à la première interface physique.

Voici les principaux arguments utilisés:

interface logique ou physique, il est obligatoire,

up active l'interface

down désactive l'interface

mtu définit l'unité de transfert des paquets

netmask affecter un masque de sous-réseau

broadcast définit l'adresse de broadcast

arp ou -arp activer ou désactiver l'utilisation du cache arp de l'interface

metric paramètre utilisé pour l'établissement des routes dynamiques, et déterminer le "coût" (nombre de sauts ou "hops") d'un chemin par le protocole RIP.

multicast activer ou non la communication avec des machines qui sont hors du réseau.

promisc ou -promisc activer ou désactiver le mode promiscuité de l'interface. En mode promiscuous, tous les paquets qui transitent sur le réseau sont reçus également par l'interface. Cela permet de mettre en place un analyseur de trame ou de protocole.

Description du résultat de la commande "ifconfig eth0":

1. eth0 Link encap:Ethernet HWaddr 00:80:C8:32:C8:1E
2. inet addr:192.168.1.1 Bcast:192.168.1.255 Mask:255.255.255.0
3. UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
4. RX packets:864 errors:0 dropped:0 overruns:0 frame:0
5. TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
6. collisions:0
7. Interrupt:10 Base address:0x6100

Explications:

Ligne 1: L'interface est de type Ethernet. La commande nous donne l'adresse MAC de l'interface.

Ligne 2 : on a l'adresse IP, celle de broadcast, celle du masque de sous-réseau

Ligne 3 : l'interface est active (UP), les modes broadcast et multicast le sont également, le MTU est de 1500 octets, le Metric de 1

Ligne 4 et 5 : RX (paquets reçus), TX (transmis), erreurs, suppressions, engorgements, collision

Mode d'utilisation:

Ce paragraphe décrit une suite de manipulation de la commande ifconfig.

Ouvrez une session en mode console sur une machine.

1 - Relevez les paramètres de votre machine à l'aide de la commande ifconfig. Si votre machine n'a qu'une interface physique, vous devriez avoir quelque chose d'équivalent à cela.

```
Lo  Link encap:Local Loopback
inet addr:127.0.0.1  Bcast:127.255.255.255  Mask:255.0.0.0
UP BROADCAST LOOPBACK RUNNING  MTU:3584  Metric:1
RX packets:146 errors:0 dropped:0 overruns:0 frame:0
TX packets:146 errors:0 dropped:0 overruns:0 carrier:0
collisions:0

eth0 Link encap:Ethernet  HWaddr 00:80:C8:32:C8:1E
inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:864 errors:0 dropped:0 overruns:0 frame:0
TX packets:654 errors:0 dropped:0 overruns:0 carrier:0
collisions:0
Interrupt:10 Base address:0x6100
```

2 - Désactivez les 2 interfaces lo et eth0

- if config lo down

- ifconfig eth0 down

3 - Taper les commandes suivantes:

- ping localhost

- ping 192.168.1.1

- telnet localhost

Aucune commande ne fonctionne, car même si la configuration IP est correcte, les interfaces sont désactivées.

4 - Activez l'interface de loopback et tapez les commandes suivantes:

- ifconfig lo up /* activation de l'interface de loopback */

- ping localhost ou telnet localhost /* ça ne marche toujours pas */

- route add 127.0.0.1 /* on ajoute une route sur l'interface de loopback */

- ping localhost ou telnet localhost /* maintenant ça marche */

- ping 192.168.1.1 /* ça ne marche pas car il manque encore une route*/

On peut déduire que :

- pour chaque interface il faudra indiquer une route au protocole.

- dans la configuration actuelle, aucun paquet ne va jusqu'à la carte, donc ne sort sur le réseau.

Voici le rôle de l'interface loopback. Elle permet de tester un programme utilisant le protocole IP, sans envoyer de paquets sur le réseau. Si vous voulez écrire une application réseau, (telnet, FTP, ou autre), vous pouvez la tester de cette façon.

5 - Activez l'interface eth0 et tapez les commandes suivantes:

- ifconfig eth0 up /* activation de l'interface */

- route add 192.168.1.1

- ifconfig /* l'information Tx/Rx de l'interface eth0 vaut 0 */

/* Aucun paquet n'est encore passé par la carte.*/

- ping 127.0.0.1

- ifconfig /* on voit que l'information Tx/Rx de lo est modifiée */

/* pas celle de eth0, on en déduit que les paquets */

/* à destination de lo ne descendent pas jusqu'à l'interface physique */

- ping 192.168.1.1 /* test d'une adresse locale */

- ifconfig /* Ici on peut faire la même remarque. Les paquets ICMP */

/* sur une interface locale, ne sortent pas sur le réseau */

/* mais ceux de l'interface lo sont modifiés*/

- ping 192.168.1.2 /* test d'une adresse distante */

- ifconfig /* Ici les paquets sont bien sortis. Les registres TX/RX de eth0 */

/* sont modifiés, mais pas ceux de lo */

6 -Réalisez les manipulations suivantes, nous allons voir le comportement de la commande ping sur les interfaces.

Sur la machine tapez la commande

192.168.1.1 ifconfig /* relevez les valeurs des registres TX/RX */

192.168.1.2 ping 192.168.1.1

192.168.1.1 ifconfig /* relevez les nouvelles valeurs des registres TX/RX */

/* il y a bien eu échange Réception et envoi de paquets*/

192.168.1.2 ping 192.168.1.3

192.168.1.1 ifconfig /* On voit que le registre Rx est modifié mais */

/* le registre Tx n'est pas modifié. La machine a bien reçu*/

/* paquet mais n'a rien renvoyé */

192.168.1.2 ping 192.168.1.2

192.168.1.2 ifconfig /* aucun registre n'est modifié, donc les paquets */

/* ne circulent pas jusqu'à l'interface physique avec un .*/

/* ping sur l'interface locale */

7 - le MTU (Message Transfert Unit) détermine l'unité de transfert des paquets.

Vous allez, sur la machine 192.168.1.1 modifier le MTU par défaut à 1500, pour le mettre à 300, avec la commande:

- ifconfig eth0 mtu 300

Sur la machine d'adresse 192.168.1.2, vous allez ouvrir une session ftp et chronométrer le temps de transfert d'un fichier de 30 MO. Relevez le temps et le nombre de paquets transmis ou reçus (commande ifconfif, flags TX/RX).

Restaurez le paramètre par défaut sur la première machine.

Refaites le même transfert et comparez les chiffres. La différence n'est pas énorme sur le temps car le volume de données est peu important. Par contre la différence sur le nombre de paquets, elle, est importante.

3.2. La commande arp

Description de la commande

La commande ARP permet de visualiser ou modifier la table du cache de l'interface. Cette table peut être statique et (ou) dynamique. Elle donne la correspondance entre une adresse IP et une adresse Ethernet.

A chaque nouvelle requête, le cache ARP de l'interface est mis à jour. Il y a un nouvel enregistrement. Cet enregistrement à une durée de vie (ttl ou Time To Leave).

Voici un exemple de cache arp, obtenu avec la commande arp -va:

? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0

Entries: 1 Skipped: 0 Found: 1

On voit l'adresse IP et l'adresse MAC correspondante. Il n'y a qu'une entrée dans la table. Voici les principales options de la commande arp:

arp -s (ajouter une entrée statique) exemple: arp -s 192.168.1.2 00:40:33:2D:B5:DD

arp -d (supprimer une entrée) exemple arp -d 192.168.1.2

Voir la page man pour les autres options.

La table ARP et le fonctionnement d'un proxy ARP.

Cela est réalisé par la configuration de tables ARP statiques.

Le proxy, est une machine qui est en interface entre un réseau et l'accès à Internet. Il fait office de passerelle et de cache à la fois.

- Passerelle, parce que tous les accès à Internet passent par le Proxy,

- Cache, parce que le Proxy conserve en mémoire cache (sur disque), une copie des pages consultées par les utilisateurs du réseau. Cela évite de télécharger à nouveau la même page sur le site d'origine, si un utilisateur revient fréquemment dessus.

Si un hôte du réseau demande l'adresse d'un noeud distant situé sur un autre réseau, et que cet hôte passe par un proxy, le proxy va renvoyer à l'hôte sa propre adresse Ethernet. Une fois cette opération réalisée, tous les paquets envoyés par l'hôte seront à destination de l'adresse Ethernet du proxy. Le proxy aura en charge de transmettre ces paquets à l'adresse effective du noeud distant.

Pour les réponses, un processus identique est mis en place. Le site consulté, ne retourne les réponses qu'au serveur proxy. Le serveur proxy se charge de ventiler les pages au bon destinataire du réseau local.

Voir, pour le fonctionnement des serveurs cache et la configuration des navigateurs avec ce type de serveur, le document sur le W3 et les scripts CGI..

Mode d'utilisation:

Attention à certaines interprétations de ce paragraphe. Il dépend de votre configuration. Soit vous êtes en réseau local avec une plage d'adresse déclarée, soit vous utilisez une carte d'accès distant.

Première partie:

1. Affichez le contenu de la table arp avec la commande arp -a,
2. Supprimez chaque ligne avec la commande arp -d @ip, où @ip est l'adresse ip de chaque hôte apparaissant dans la table,
3. La commande arp -a ne devrait plus afficher de ligne,
4. Faites un ping, sur une station du réseau local,
5. arp -a, affiche la nouvelle entrée de la table,
6. Ouvrez une session sur Internet, puis ouvrez une session ftp anonyme sur un serveur distant en utilisant le nom, par exemple ftp.cdrom.com. Utilisez une adresse que vous n'avez jamais utilisée, supprimez également tout gestionnaire de cache.
7. Affichez le nouveau contenu de la table avec arp-a. Le cache arp ne contient pas l'adresse Ethernet du site distant, mais celle de la passerelle par défaut. Cela signifie que le client n'a pas à connaître les adresses Ethernet des hôtes étrangers au réseau local, mais uniquement l'adresse de la passerelle. Les paquets sont ensuite pris en charge par les routeurs.
8. Refaites une tentative sur le site choisi précédemment. Le temps d'ouverture de session est normalement plus court. Cela est justifié, car les serveurs de noms ont maintenant dans leur cache la correspondance entre le nom et l'adresse IP.

Deuxième partie:

La commande ARP permet de diagnostiquer un dysfonctionnement quand une machine prend l'adresse IP d'une autre machine.

1. Sur la machine 192.168.1.1, faites un ping sur 2 hôtes du réseau 192.168.1.2 et 192.168.1.3,

2. À l'aide de la commande arp, relevez les adresses MAC de ces noeuds,
3. Modifiez l'adresse IP de la machine 192.168.1.2 en 192.168.1.3
4. relancez les 2 machines en vous arrangeant pour que la machine dont vous avez modifié l'adresse ait redémarré la première,
5. Sur la machine d'adresse 192.168.1.1, remettez à jour les tables arp,
6. Quel est le contenu, après cela de la table arp ?

Conclusion : Vous allez avoir un conflit d'adresses. Vous allez pouvoir le détecter avec la commande arp. Autre problème, si vous faites un telnet sur 192.168.1.3, il y a de fortes chances pour que ce soit la machine qui était d'adresse 192.168.1.2, qui vous ouvre la session. Nous sommes (par une action volontaire bien sûr) arrivés à mettre la pagaille sur un réseau de 3 postes. Cette pagaille pourrait tourner vite au chaos sur un grand réseau, d'où la nécessité pour un administrateur de faire preuve d'une grande rigueur.

Où en suis-je ?

Exercice 1:

Vous êtes sur un réseau d'adresse 192.168.1.0 avec une interface d'adresse MAC 00:40:33:2D:B5:DD,

Vous n'avez aucun fichier host sur votre machine,

Il n'y a pas de DNS,

La passerelle par défaut est 192.168.1.9

Vous faites un "ping 195.6.2.3" qui a une interface d'adresse MAC 00:45:2D:33:C2 est localisée sur Internet

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.168.1.2) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp? (192.168.1.2) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier host, ou DNS pour réaliser l'opération ping demandée

F - Il n’est pas possible dans la configuration actuelle d’atteindre l’hôte 195.6.2.3

Réponse F, car la plage d’adresse 192.168.1.1 à 192.168.1.254 n’est pas routée sur l’Internet, sinon vous auriez l’adresse de la passerelle par défaut dans le cache arp.

Exercice 2:

Vous êtes sur un réseau d’adresse 192.5.1.0 avec une interface d’adresse MAC 00:40:33:2D:B5:DD,

Vous n’avez aucun fichier host sur votre machine,

Il n’y a pas de DNS,

La passerelle par défaut est 192.5.1.9

Vous faites un "ping www.existe.org" dont l’adresse ip est 195.6.2.3, et qui a une interface d’adresse MAC 00:45:2D:33:C2

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp? (195.6.2.3) at 00: 00:45:2D:33:C2 [ether] on eth0

E - Il faut un fichier host, ou DNS pour réaliser l’opération ping demandée

F - Il n’est pas possible dans la configuration actuelle d’atteindre l’hôte 195.6.2.3

Réponse E, car la résolution de noms ne peut être effectuée

Exercice 3:

Vous êtes sur un réseau d’adresse 192.5.1.0, sur une machine d’adresse 192.5.1.1, et une interface d’adresse MAC 00:40:33:2D:B5:DD,

Vous n’avez aucun fichier host sur votre machine,

Il n’y a pas de DNS,

La passerelle par défaut est 192.5.1.9, d’adresse MAC 09:44:3C:DA:3C:04

Vous faites un "ping 195.6.2.3", et qui a une interface d’adresse MAC 00:45:2D:33:C2

Le réseau fonctionne parfaitement et tout est parfaitement configuré

Cochez la bonne réponse:

A - On a dans la table arp ? (192.5.1.0) at 00:40:33:2D:B5:DD [ether] on eth0

B - On a dans la table arp? (192.5.1.0) at 00:45:2D:33:C2 [ether] on eth0

C - On a dans la table arp? (195.6.2.3) at 00:40:33:2D:B5:DD [ether] on eth0

D - On a dans la table arp? (192.5.1.9) at 09:44:3C:DA:3C:04 [ether] on eth0

E - Il faut un fichier host, ou DNS pour réaliser l’opération ping demandée

F - Il n’est pas possible dans la configuration actuelle d’atteindre l’hôte 195.6.2.3

Réponse D, l’hôte a bien été trouvé, la table arp a été mise à jour avec l’adresse ip de la passerelle par défaut et son adresse Ethernet.

3.3. La commande route

La commande route a déjà été entrevue un peu plus haut, avec la commande ifconfig. Le routage définit le chemin emprunté par les paquets entre son point de départ et son point d’arrivée. Cette commande permet également la configuration de pc, de switchs de routeurs.

Il existe 2 types de routages:

- le routage statique

- le routage dynamique.

Le routage statique consiste à imposer aux paquets la route à suivre.

Le routage dynamique met en oeuvre des algorithmes, qui permettent aux routeurs d'ajuster les tables de routage en fonction de leur connaissance de la topologie du réseau. Cette actualisation est réalisée par la réception des messages reçus des noeuds (routeurs) adjacents.

Le routage dynamique permet d'avoir des routes toujours optimisées, en fonction de l'état du réseau (nouveaux routeurs, engorgements, pannes)

On combine en général le routage statique sur les réseaux locaux au routage dynamique sur les réseaux importants ou étendus.

Un administrateur qui dispose par exemple de 2 routeurs sur un réseau, peut équilibrer la charge en répartissant un partie du flux sur un port avec une route, et une autre partie sur le deuxième routeur.

Exemple de table de routage:

```
Kernel IP routing table
Destination  Gateway  Genmask   Flags  Metric  Ref Use  Iface
192.168.1.0  *   255.255.255. 0U     0       0   2    eth0
127.0.0.0  *   255.0.0.0    U      0       0   2    lo
default   192.168.1.9  0.0.0.0    UG     0       0   10   eth0
```

Commentaire généraux:

Destination : Adresse de destination de la route

Gateway: Adresse ip de la passerelle pour atteindre la route, * sinon

Genmask : Masque à utiliser.

Flags : Indicateur d'état (U - Up, H - Host - G - Gateway, D - Dynamic, M - Modified)

Metric : Coût métrique de la route (0 par défaut)

Ref : Nombre de routes qui dépendent de celle-ci

Use : Nombre d'utilisation dans la table de routage

Iface : Interface eth0, eth1, lo

Commentaire sur la 3ème ligne:

Cette ligne signifie que pour atteindre tous les réseaux inconnus, la route par défaut porte l'adresse 192.168.1.9. C'est la passerelle par défaut, d'où le sigle UG, G pour gateway.

Ajout ou suppression d'une route:

route add [net | host] addr [gw passerelle] [métric coût] [netmask masque] [dev interface]

- *net ou host* indique l'adresse de réseau ou de l'hôte pour lequel on établit une route,

- adresse de destination,

- adresse de la passerelle,

- valeur métrique de la route,

- masque de la route à ajouter,

- interface réseau à qui on associe la route.

Exemples:

route add 127.0.0.1 lo /* ajoute une route pour l'adresse 127.0.0.1 sur l'interface lo */

route add -net 192.168.2.0 eth0 /* ajoute une route pour le réseau 192.168.2.0 sur l'interface eth0 */

route add saturne.foo.org /* ajoute une route pour la machine machin sur l'interface eth0 */

route add default gw ariane /* ajoute ariane comme route par défaut pour la machine locale */

/* ariane est le nom d'hôte d'un routeur ou d'une passerelle */

/* gw est un mot réservé */

route add duschmoll netmask 255.255.255.192

/* Encore un qui a créé des sous réseaux., Il s'agit ici d'une classe c */

/* avec 2 sous réseaux, il faut indiquer le masque. */

Suppression d'une route:

route del -net 192.168.1.0

route del -net toutbet-net

Attention: si on utilise des noms de réseau ou des noms d'hôtes, il faut qu'à ces noms soient associés les adresses de réseau ou des adresses ip dans le fichier /etc/networks pour les réseaux, et /etc/hosts ou DNS pour les noms d'hôtes.

Vous pouvez également voir l'atelier sur la mise en place d'un routeur logiciel.

Petite étude de cas:

Première partie - réalisation d'une maquette:

On dispose de 2 réseaux (A et B) reliés par une passerelle. Le réseau A est également relié à Internet par un routeur. Le réseau A dispose d'un serveur de noms. Chaque réseau a deux machines.

```
Réseau  Nom du réseau Machine Nom des machines
A metaux-net 192.3.2.2 platine
192.3.2.3 uranium
192.3.2.4   mercure (serveur de noms)

B roches-net 130.2.0.2 quartz
130.2.0.3 silex
```

La passerelle entre le réseau A et B à 2 interfaces:

- eth0 192.3.2.1

- eth1 130.2.0.1

Le réseau A, a une passerelle par défaut pour Internet 130.2.0.9, qui est l'interface d'un autre routeur.

On veut:

- que les stations de chaque réseau puissent accéder à Internet,

- que les stations de chaque réseau puissent communiquer entre-elles,

- que les stations du réseau B, utilisent le serveur de noms le moins possible.

On demande:

1 - d'expliquer comment seront configurés les postes du réseau B,

2 - de donner la configuration des fichiers suivants pour chaque machine (hosts, resolv.conf, fichier de configuration de carte).

3 - de donner la liste des routes à mettre:

- sur les postes du réseau B,

- sur les postes du réseau A,

- sur la passerelle qui relie les 2 réseaux,

- sur le routeur du réseau A.

3.4. La commande netstat

La commande netstat, permet de tester la configuration du réseau, visualiser l'état des connexions, établir des statistiques, notamment pour surveiller les serveurs.

Liste des paramètres utilisables avec netstat:

Sans argument, donne l'état des connexions,

-*a* afficher toutes les informations sur l'état des connexions,

-*i* affichage des statistiques,

-*c* rafraîchissement périodique de l'état du réseau,

-*n* affichage des informations en mode numérique sur l'état des connexions,

-*r* affichage des tables de routage,

-*t* informations sur les sockets TCP

-*u* informations sur les sockets UDP.

Etat des connexions réseau avec netstat, dont voici un exemple:

```
 Proto  Recv-Q  Send-Q  Local Address   Foreign Address  State
 Tcp  0   126   uranus.planete.n:telnet 192.168.1.2:1037 ESTABLISHED
 Udp    0       0       uranus.plan:netbios-dgm  *:*
 Udp    0       0       uranus.plane:netbios-ns  *:*

Active UNIX domain sockets (w/o servers)
 Proto  RefCnt  Flags        Type        State         I-Node Path
 unix   2       [ ]          STREAM      1990     /dev/log
```

```
unix   2       [ ]         STREAM CONNECTED 1989
unix   1       [ ]         DGRAM        1955
```

Explications sur la première partie qui affiche l'état des connexions:

Proto : Protocole utilisé

Recv-q : Nbre de bits en réception pour ce socket

Send-q : Nbre de bits envoyés

LocalAdress : Nom d'hôte local et port

ForeignAdress : Nom d'hôte distant et port

State : Etat de la connexion

Le champ state peut prendre les valeurs suivantes:

Established : Connexion établie

Syn snet : Le socket essaie de se connecter

Syn recv : La connexion s'initialise

Fin wait1 : Le socket a été fermé

Fin wait2 : La connexion a été fermée

Closed : Le socket n'est pas utilisé

Close wait : L'hôte distant a fermé la connexion; Fermeture locale en attente.

Last ack : Attente de confirmation de la fermeture de la connexion distante

Listen : Ecoute en attendant une connexion externe.

Unknown : Etat du socket inconnu

Explications sur la deuxième partie qui affiche l'état des sockets (IPC - Inter Processus Communication) actifs:

Proto : Protocole, en général UNIX,

Refcnt : Nombre de processus associés au socket

Type : Mode d'accès datagramme (DGRAM), flux orienté connexion (STREAM), brut (RAW), livraison fiable des messages (RDM)

State : Free, Listening, Unconnected, connecting, disconnecting, unknown

Path : Chemin utilisé par les processus pour utiliser le socket.

Affichage et état des tables de routage avec netstat: netstat -nr ou netstat -r

```
Kernel IP routing table
Destination   Gateway   Genmask         Flags   MSS    Window  irtt  Iface
192.168.1.0   *         255.255.255.0   U       1500   0       0     eth0
127.0.0.0     *         255.0.0.0       U       3584   0       0     lo
```

Explications sur la commande netstat -r

Destination: Adresse vers laquelle sont destinés les paquets

Gateway : Passerelle utilisée, * sinon

Flags : G la route utilise une passerelle, U l'interface est active, H on ne peut joindre qu'un simple hôte par cette route)

Iface : Interface sur laquelle est positionnée la route.

Affichage de statistiques avec netstat -i

```
Iface MTU  Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags
Lo    3584 0   89    0      0      0      89    0      0      0      BLRU
eth0  1500 0   215   0      0      0      210   0      0      0      BRU
```

Explications sur la commande netstat -i

RX-OK et TX-OK rendent compte du nombre de paquets reçus ou émis,

RX-ERR ou TX-ERR nombre de paquets reçus ou transmis avec erreur,

RX-DRP ou TX-DRP nombre de paquets éliminés,

RX-OVR ou TX-OVR recouvrement, donc perdus à cause d'un débit trop important.

Les Flags (B adresse de diffusion, L interface de loopback, M tous les paquets sont reçus, O arp est hors service, P connexion point à point, R interface en fonctionnement, U interface en service)

Exercices:

On donne les résultats de 3 commandes netstat ci-dessous, extraites de la même machine:

$ netstat -nr

Kernel IP routing table

Destination Gateway Genmask Flags MSS Window irtt Iface

198.5.203.0 0.0.0.0 255.255.255.0 U 1500 0 0 eth0

127.0.0.0 0.0.0.0 255.0.0.0 U 3584 0 0 lo

0.0.0.0 198.5.203.3 0.0.0.0 UG 1500 0 0 eth0

$netstat

Active Internet connections (w/o servers)

Proto Recv-Q Send-Q Local Address Foreign Address State

Tcp 0 127 uranus.toutbet:telnet 194.206.6.143:1027 ESTABLISHED

$ netstat -i

Iface MTU Met RX-OK RX-ERR RX-DRP RX-OVR TX-OK TX-ERR TX-DRP TX-OVR Flags

Lo 3584 0 764 0 0 764 89 0 0 0 BLRU

eth0 1500 0 410856 0 0 33286 210 0 0 0 BRU

On demande:

1. Quels sont les noms et adresse de la machine consultée ?

2. Quel type de session est-elle en train de supporter ?
3. A quoi correspond l'adresse 198.5.203.3 ?
4. Pourquoi une interface porte-t-elle les Flags BLRU et l'autre BRU ?
5. Quelle est la taille des paquets utilisée par la passerelle par défaut ?

3.5. La commande traceroute

La commande traceroute, permet d'afficher le chemin parcouru par un paquet pour arriver à destination. Cette commande est importante, car elle permet d'équilibrer la charge d'un réseau, en optimisant les routes.

Voici le résultat de la commande "traceroute www.nat.f"r, tapée depuis ma machine.

```
traceroute to sancy.nat.fr (212.208.83.2), 30 hops max, 40 byte packets
 1  195.5.203.9 (195.5.203.9)  1.363 ms  1.259 ms  1.270 ms
 2  194.79.184.33 (194.79.184.33)  25.078 ms  25.120 ms  25.085 ms
 3  194.79.128.21 (194.79.128.21)  88.915 ms  101.191 ms  88.571 ms
 4  cisco-eth0.frontal-gw.internext.fr (194.79.190.126)  124.796 ms[]
 5  sfinx-paris.remote-gw.internext.fr (194.79.190.250)  100.180 ms[]
 6  Internetway.gix-paris.ft.NET (194.68.129.236)  98.471 ms       []
 7  513.HSSI0-513.BACK1.PAR1.inetway.NET (194.98.1.214)  137.196 ms[]
 8  602.HSSI6-602.BACK1.NAN1.inetway.NET (194.98.1.194)  101.129 ms[]
 9  FE6-0.BORD1.NAN1.inetway.NET (194.53.76.228)  105.110 ms       []
10  194.98.81.21 (194.98.81.21)  175.933 ms  152.779 ms  128.618 ms[]
11  sancy.nat.fr (212.208.83.2)  211.387 ms  162.559 ms  151.385 ms[]
```

Explications:

Ligne 0 : le programme signale qu'il n'affichera que les 30 premiers sauts, et que la machine www du domaine nat.fr, porte le nom effectif de "sancy", dans la base d'annuaire du DNS du domaine nat.fr. Cette machine porte l'adresse IP 212.208.83.2. Pour chaque tronçon, on a également le temps maximum, moyen et minimum de parcours du tronçon.

Ensuite, on a pour chaque ligne, l'adresse du routeur que le paquet a traversé pour passer sur le réseau suivant.

Ligne 4 et 5, le paquet a traversé 2 routeurs sur le même réseau 194.79.190.

Ligne 4, 5, 6, 7, 8, 9, 11, on voit que les routeurs ont un enregistrement de type A dans les serveurs de noms, puisqu'on voit les noms affichés.

Conclusion : Depuis ma machine, chaque requête HTTP passe par 11 routeurs pour accéder au serveur www.nat.fr.

L'accès sur cet exemple est réalisé sur Internet. Un administrateur, responsable d'un réseau d'entreprise sur lequel il y a de nombreux routeurs, peut, avec cet outil, diagnostiquer les routes et temps de routage. Il peut ainsi optimiser les trajets et temps de réponse.

3.6. La commande dig

La commande dig remplace ce qui était la commande nslookup. Cette commande sert à diagnostiquer des dysfonctionnements dans la résolution de nom. (Service DNS).

Utilisation simple de dig:

```
$ dig any freenix.org

; <<>> DiG 9.2.2 <<>> any freenix.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 21163
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 3, ADDITIONAL: 3

;; QUESTION SECTION:
;freenix.org.                   IN      ANY

;; ANSWER SECTION:
freenix.org.            92341   IN      SOA     ns2.freenix.org.\
                                         hostmaster.freenix.org.\
                                                     2003042501\
                                                          21600\
                                                           7200\
                                                        3600000\
                                                         259200\
freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.

;; AUTHORITY SECTION:
freenix.org.            117930  IN      NS      ns2.freenix.fr.
freenix.org.            117930  IN      NS      ns.frmug.org.
freenix.org.            117930  IN      NS      ns6.gandi.net.

;; ADDITIONAL SECTION:
ns2.freenix.fr.         16778   IN      A       194.117.194.82
ns.frmug.org.           40974   IN      A       193.56.58.113
ns6.gandi.net.          259119  IN      A       80.67.173.196

;; Query time: 197 msec
;; SERVER: 213.36.80.1#53(213.36.80.1)
;; WHEN: Tue May 27 15:16:23 2003
;; MSG SIZE  rcvd: 248
```

retourne les informations sur le domaine concerné.

Il est ensuite possible d'intérroger sur tout type d'enregistrement : SOA, MX, A, CNAME, PTR...

3.7. La comande host

La commande host interroge les serveurs de coms. Elle peut par exemple être utilisée pour détecter des dysfonctionnement sur un réseau (serveurs hors services). Attention, n’utilisez pas cette commande sur des réseaux dont vous n’avez pas l’administration.

Les éditeurs joe et Émacs

Commandes de base pour pouvoir modifier les fichiers de configuration.

1. Les éditeurs de texte Emacs et Joe

1.1. Présentation

Ce document donne les principales commandes qui permettent de commencer à utiliser un éditeur sous Linux. Emacs et Joe sont des éditeurs très utilisés sous Linux. Ils prennent peu à peu le pas sur VI (prononcer vi aïe). Sous Xwindow vous pouvez également utiliser Xemacs. Ces éditeurs sont normalement installés avec l'installation de Linux. Si cela n'est pas le cas, il vous faudra les installer ultérieurement. Les éditeurs sont les principaux outils utilisés pour la création de scripts ou de programmes sources. Leur principale différence avec un traitement de texte est qu'ils ne mettent aucun caractère de contrôle dans le document. Vous n'avez pas la possibilité de mettre en gras, italique, souligné. Pour installer par exemple l'éditeur joe, copiez le programme joe-2.8-9.i386.rpm de votre CD ROM sur le disque dur dans /temp. Installez-le avec la commande "rpm -i joe-2.8-9.i386.rpm". Le programme joe est maintenant installé dans le répertoire /usr/bin. Vous pouvez l'utiliser en tapant joe.

1.2. L'éditeur Joe

```
Pour opbtenir de l'aide CTRL h

Les commandes de base

Rechercher CTRL k f
Rechercher suivant CTRL k l

Copier un block
Début de block CTRL k b
Fin de block CTRL k k
Copier le block CTRL k c
Déplacer le block CTRL k m
Supprimer le block CTRL k y


Ecran précédent CTRL u
Ecran suivant CTRL v
Début de document CTRL k u
Fin de document CTRL k v

Début de ligne CTRL a
```

```
Fin de ligne  CTRL e

Sauvegarder et quitter CTRL k x
Sauvegarder CTRL k d
Lire un fichier CTRL k e
Insérer un fichier CTRL k r

Accéder au shell  CTRL k z (taper fg pour revenir)

Quitter  CTRL x c
```

1.3. L’éditeur Emacs

```
Notation des touches :
CTRL : signifie Ctrl
META : signifie Alt
ESC : signifie ECHAP
SHT : signifie SHIFT
RET : return
SPB : signifie barre d’espace

Les commandes de base :
Lancement de Emacs :
emacs : lancement avec un fichier vide
emacs NomFichier : édite le Fichier de nom NomFichier

Action   Touches

Accéder à l’aide  CTRL h
Répertoire : liste    CTRL x  CTRL d
Annuler Cmd en cours  CTRL g
Annuler cmd précédente    CTRL x  u
Annuler modifications     ESC  ~
Curseur End    CTRL e
Curseur Home  CTRL a
Reculer d’un caractère   CTRL b
Avancer d’un caractère    CTRL f
Défilement PgDn   CTRL v
Défilement PgUp   ESC  v
Effacer caractère droite CTRL d
Effacer fin de ligne  CTRL k
Fichier : charger     CTRL x  CTRL f
Fichier : insérer     CTRL x  CTRL i
Fichier : sauver  CTRL x  CTRL s
Fichier : (re)nommer  CTRL x CTRL w nom
Positionnement haut   ESC  <
Positionnement bas    ESC  >
Rechercher    CTRL s
Remplacer     ESC  %
Bloc: marque debut     CTRL SPB
Coller region (paste)     CTRL y
Copier region (copy)  ESC  w
Couper region (cut)   CTRL w
Aller à la ligne ..   ESC  x
Quitter   CTRL x  CTRL c
```

Annuler une commande CTRL g

Gestion des fenêtres et des Buffers

Liste des buffers CTRL x CTRL b
Changer de fenêtre CTRL x o
Maximiser la fenêtre courante CTRL x 1

Installation d’un serveur Telnet et FTP

1. Résumé

Le document décrit l’installation d’un service de transfert de fichier, la configuration du démon inetd et quelques premiers aspects touchant à la sécurité des services sous GNU/Linux.

Mots clés : Telnet, FTP, ssh, sftp, scp, TCP-Wrapper

1.1. Description et objectifs de la séquence

Vous devriez à la fin pouvoir :

- utiliser le service FTP du serveur à partir d’un client quelconque du réseau
- bénéficier du service ftp anonyme ou authentifié,
- pouvoir filtrer l’accès provenant de tout ou partie du réseau avec TCP-Wrapper.

1.2. Présentation des concepts importants

1) Telnet:

Telnet est un protocole qui permet l’émulation de terminal VTx à distance sur un serveur Unix/Linux.

2) FTP:

FTP est un protocole de communication qui permet le transfert de fichier entre plusieurs machines.

3) Le daemon inetd:

Toute application fonctionnant sous TCP/IP est basée sur le modèle client/serveur. Par exemple quelqu’un se connectant grâce à telnet à un hôte distant « active » chez l’hôte le service serveur telnetd.

Chaque serveur est sur une machine en attente d’une connexion sur un port particulier. Dans les premières versions d’Unix-TCP/IP chaque application (telnet, ftp,...) avait son propre serveur qui était lancé au démarrage de chaque machine comme un "daemon". Cette stratégie encombrait inutilement la table des processus (autant de serveurs que de services). Ces services sont dits fonctionnant en mode « autonome » ou « standalone ».

Le daemon INETD est un « super » serveur, à l'écoute sur plusieurs ports et qui se charge de recevoir les demandes de connexion de plusieurs clients (telnet, ftp,...) et de lancer le serveur correspondant à la demande. A son démarrage il consulte les fichiers:

- /etc/services qui contient la liste générale des services TCP/IP avec leur numéro de port et le protocole de transport associé.

- /etc/inetd.conf qui contient la liste des services activés sur une machine donnée

Dans les distributions récentes (Mandrake 8.x, RedHat 7.x...), inetd a été remplacé par xinetd. Le principe est très similaire, à la seule différence que, dans /etc/etc/xinetd.d, chaque service (telnet, ftp, pop3...) dispose de son propre fichier de configuration.

Certains services utilisable avec inetd ou xinetd comme telnet, ftp, pop3... sont difficilement sécurisables car les mots de passe transitent en clair sur le réseau. Ce problème sera vu ultérieurement avec les TPs sur la métrologie. Si ces services sont utilisables en l'état sur des petits réseaux isolés, il faudra éviter de les utiliser sur des réseaux reliés à Internet ou dans des environnements peu sûrs. Cependant, la tendance est au cryptage de ces services, grâce à SSL notament. Il existe une version sécurisée de telnet, nommée telnet-ssl.

1.3. Extrait de /etc/services :

/etc/services :

```
ftp  21/tcp
telnet  23/tcp
smtp  25/tcp  mail
pop3  110/tcp  # Post Office
```

etc...

1.4. Extrait de /etc/inetd.conf

```
ftp      stream  tcp     nowait  root    /usr/sbin/ftpd          ftpd
#shell   stream  tcp     nowait  root    /usr/sbin/rshd          rshd
#login   stream  tcp     nowait  root    /usr/sbin/rlogind       rlogind
#exec    stream  tcp     nowait  root    /usr/sbin/rexecd        rexecd
```

Ici, il n'y a que le service ftp qui est activé par le serveur inetd. Les autres lignes sont en commentaires.

Ces services sont dits fonctionnant en mode « parallèle ».

1.5. Confi guration avec xinetd

Le principe est similaire, à la différence que vous avez un fichier de configuration global "/etc/xinetd.conf", et un fichier de configuration par service, en général dans "/etc/xinetd.d".

```
#
# Le fichier xinetd.conf
#
# Some defaults, and include /etc/xinetd.d/

defaults
{
        instances               = 60
        log_type                = SYSLOG authpriv
        log_on_success          = HOST PID
        log_on_failure          = HOST
        cps                     = 25 30
}

includedir /etc/xinetd.d
```

Le fichier /etc/xinetd.d/wu-ftpd

```
# default: on
# description: The wu-ftpd FTP server serves FTP connections. It uses \
#       normal, unencrypted usernames and passwords for authentication.
service ftp
{
        disable = no
        socket_type             = stream
        wait                    = no
        user                    = root
        server                  = /usr/sbin/in.ftpd
        server_args             = -l -a
        log_on_success          += DURATION USERID
        log_on_failure          += USERID
        nice                    = 10
}
```

Le paramètre "disable", permet d’activer/désactiver le service.

le programme "in.ftpd", indique bien que le service est pris en charge par TCPWrapper.

Les commentaires en haut du fichier indiquent que ce service ne prend pas en charge l’encryptage des mots de passe.

1.6. TCP-Wrapper

5) TCPWrapper:

TCP-Wrapper est un outil de sécurité réseau qui permet de contrôler les accès, les tentatives de connexion sur une machine donnée. Il permet à tout instant de savoir (par journalisation syslogd) qui essaie d’accéder sur un ordinateur mais également de filtrer les accès. On peut par exemple sur une machine A interdire les connexions telnet venant d’une machine B tout en autorisant les connexions FTP venant de cette même machine B.

Principe de fonctionnement:

Exemple: Si inetd reçoit une demande de connexion sur le port 23 il va lancer telnetd.

Tcpwrapper sert d'enveloppe. Il vient « s'intercaler » entre le daemon inetd et le serveur à démarrer. Quand une demande de service TCP/IP (en réalité TCP ou UDP) arrive sur un port donné, inetd va lancer TCPD (daemon correspondant à Tcpwrapper) au lieu d'activer directement le service demandé (telnetd, ftpd, pop3...).

Tcpd prend en charge la requête et met en place ses mécanismes de contrôle. Il peut par exemple vérifier que les accès depuis la machine cliente sont autorisés. Une fois le traitement terminé il va (s'il y a autorisation) lancer son propre service in.telnetd, in.ftpd, in.imapd....

1.7. Eléments de confi guration

Sous Linux tcpd est installé par défaut. On peut voir en consultant le fichier /etc/inetd.conf comment inetd active tcpd.

1.7.1. Extrait de /etc/inetd.conf

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd -l -a
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
```

1.7.2. TCP Wrapper

L'administrateur réseau va pouvoir utiliser 2 fichiers: /etc/hosts.allow et /etc/hosts.deny pour filtrer les accès à sa machine.

/etc/hosts.deny: on indique dans ce fichier les services et les hôtes pour lesquels l'accès est interdit.

/etc/hosts.allow: on indique dans ce fichier les services et les hôtes pour lesquels l'accès est autorisé.

Exemple:

```
# Fichier /etc/hosts.deny
# interdit tous les accès ftp à la machine
in.ftpd:ALL

# Fichier /etc/hosts.allow
# autorise les accès ftp venant de cli1
in.ftpd :cli1.archinet.edu
```

TCP-Wrapper utilise l'algorithme suivant :

```
Si une règle est applicable dans hosts.allow, alors cette règle est appliquée, sinon,
Si une règle est applicable dans hosts.deny alors cette règle est appliquée, sinon,
```

```
l’accès est autorisé.
```

Ce mode de fonctionnement induit la stratégie de sécurité à adopter :

1. décrire toutes les règles pour les couples (services/clients) qui sont autorisés,
2. interdire systématiquement tout le reste. Mettre par défaut ALL:ALL dans hosts.deny.

Les tentatives d’accès depuis des machines extérieures sont toutes enregistrées dans des fichiers particuliers. Ces enregistrements sont effectués par le processus syslogd qui à son démarrage lit le fichier /etc/syslog.conf pour trouver dans quel(s) fichier(s) il doit enregistrer les différentes tentatives d’accès.

1.8. Extrait de /etc/syslog.conf

```
# Log anything (except mail) of level info or higher.
# Don’t log private authentication messages

# The authpriv file has restricted access.
authpriv.*                                              /var/log/auth.log
auth,authpriv.none;                         /var/log/syslog
```

1.9. Extrait de /var/log/syslog

```
Feb  3 18:02:52 ns1 ftpd[1051]: FTP session closed
Feb  3 18:03:31 ns1 syslogd 1.3-3: restart.
Feb  3 18:07:34 ns1 in.ftpd[1057]: refused connect from cli1.archinet.edu
Feb  3 18:07:46 ns1 in.ftpd[1058]: connect from ns1.archinet.edu
Feb  3 18:10:57 ns1 login[1063]: LOGIN ON ttyp3 BY mlx FROM puce
```

Remarques:

La commande kill -HUP pid de sysklogd permet de redémarrer ce processus avec prise en compte des paramètres se trouvant dans /etc/syslog.conf. Il est aussi possible d’invoquer le script de lancement du service en lui passant l’argument restart : /etc/init.d/syslogd restart

1.10. Processus d’installation et de configuration

Ouvrez le fichier /etc/inetd.conf, vérifiez que les lignes qui activent les démons telnet et ftp sont décommentées.

Vérifier qu’il n’y a aucune règle active dans les fichiers /etc/hosts.allow et /etc/hosts.deny (mettez les règles éventuelles en commentaire)

Pour l’activer manuellement utilisez la commande : /etc/init.d/inetd stop | start

1.11. Procédure de tests

1 - Créez un compte d’utilisateur.

2 - Sur la console, ouvrez une session sous le compte root.

3 - Vous devez pouvoir utiliser les commandes :

```
«#> ftp localhost » ou « ftp ‘hostname‘ » en vous authentifiant avec le compte que vous avez créé
«#> telnet localhost » ou « telnet ‘hostname‘ » en utilisant le compte que vous avez créé.
où ‘hostname‘ indique le nom d’hôte de votre machine.
```

Si ces commandes fonctionnent sur le serveur, réaliser les opérations à partir d’un client distant.

Vous pouvez vérifier le fonctionnement de « tcpwraper »

1 - Interdisez tout dans le fichier /etc/hosts.deny (mettre ALL:ALL à la fin du fichier). Attention, mettez plusieurs retours chariots (CR/LF) à la fin du fichier sinon la dernière ligne n’est pas lue. Vous avez des exemples dans "man 5 hosts_access" ou man "hosts.allow".

2 - Vérifiez que l’accès ftp est maintenant refusé, vérifiez les messages dans /var/log/syslog et /var/log/auth.log

Vous devriez voir, dans les fichiers de log (journaux) les demandes ftp rejetées par TCPWrapper.

Remettez le fichier hosts.deny dans son état initial.

1.12. Problèmes rencontrés

Q - je ne peux pas accéder au serveur en utilisant le compte root.

R - Vous pouvez réaliser cette opération sur la console, mais par mesure de sécurité cette opération n’est pas possible à distance. Avec telnet, ouvrez une session sur un compte d’utilisateur standard, puis la commande « su ».

Q - Je n’arrive pas ouvrir de session Telnet ou FTP.

R - Vérifier le fichier de configuration « /etc/inetd.conf », puis que le serveur inet est bien lancé.

Q - J’ai modifié les fichiers host.allow et host.deny. Les modifications n’ont pas l’air d’être prises en compte.

R - Vérifiez la syntaxe des instructions utilisées dans ces fichiers, normalement la modification des règles est prise en compte dynamiquement sans avoir besoin de relancer le service. Insérez quelques lignes vides à la fin de ces fichiers.

Attention : les services telnet et ftp n’offrent aucune solution de sécurité sur les réseaux (transmission des données en clair). Sur un réseau qui n’est pas sûr vous ne devrez pas utiliser ces services.

Il y a d’autres fichiers de configuration qui permettent de sécuriser le service FTP. Ces fichiers, dans /etc, sont indépendants de TCP Wrapper. Regardez ftpaccess, ftpgroup, ftphosts, ftpusers et leurs pages de manuel. Avec ftpusers, vous pouvez autoriser/interdire l’accès pour un compte en particulier.

Sources de documentation complémentaires

Les pages du manuel de TCPWrapper. man syslog.conf ou man syslogd pour plus de renseignements.

2. Application

2.1. Quelques remarques

- Relevez les ports utilisés par les services telnet, ftp, pop3, dns, smtp, http.
- Installez et testez les services telnet et ftp à partir de votre poste puis à partir d’un autre poste. Utilisez les traces de journaux pour identifier les problèmes.
- Utilisez TcpWrapper pour autoriser/interdire le service telnet, le service ftp, tous les services. Vous testerez l’accès à partir de votre poste, d’un autre poste.

Attention : Pensez à relancer un service serveur chaque fois que vous avez modifié son fichier de configuration, ceci est vrai pour tous les services et ne sera plus répété. En général utilisez la manipulation suivante "/etc/init.d/NomDuService start | stop | status | restart"

Il est possible que le client ftp soit remplacé par un autre programme comme "lftp" par exemple. C’est ce programme qui sera utilisé dans le TP, vous adapterez si vous utilisez autre chose. Fondamentalement ça ne changera rien, mais lftp est beaucoup plus riche fonctionnellement que les clients ftp standard. Il supporte 6 méthodes d’accès ftp, ftps, http, https, hftp et fish.

La freeduc-sup est configurée pour ne pas supporter les tansations et protocoles "non-sûrs". Si vous utilisez un client autre comme une knoppix par exemple,vous devrez installer le support ssl.

```
apt-get install telnetd-ssl
```

2.2. Confi guration de telnet

1. Relevez le port utilisé par telnet dans le fichier /etc/services
2. Décommentez la ligne qui concerne telnet dans /etc/inetd.conf et relancez le service.
3. Vérifiez que le port est bien ouvert avec la commande netstat :

```
#> netstat -atup | grep LISTEN
```

4. Vérifiez que rien dans TCP-Wrapper n’interdise l’accès au service telnet.

   ```
   # Mettre dans /etc/hosts.allow
   ALL:ALL
   ```

   Testez l’accès au service telnet.

2.3. Configuration de TCP-Wrapper

1. Interdisez tous les accès dans TCP-Wrapper, testez.

   ```
   # Commentez toutes les lignes dans /etc/hosts.allow
   # Mettez dans /etc/hosts.deny
   ALL:ALL
   ```

2. En vous aidant des exemples donnés dans "man hosts.allow", autorisez l’accès pour une machine du réseau sur le service telnet, interdisez pour toutes les autres, testez.

2.4. Test de l’accès ftp authentifié

L’accès authentifié est simple à mettre en oeuvre.

1. Relevez les ports utilisés par ftp dans /etc/services.
2. Activez le service dans inetd.conf et lancer le service.
3. Vérifier que le port est bien ouvert avec la commande netstat
4. Vérifier que rien n’interdit l’accès au service ftp dans les fichiers hosts.allow et hosts.deny
5. Faites un test en utilisant un compte système existant, par exemple "lftp loalhost -u mlx" si mlx est votre compte.

Normalement c’est terminé, tout doit fonctionner. Si cela ne fonctionne pas, vérifiez que vous n’avez rien oublié, vérifiez aussi les fichiers de log (/var/log/daemon, /var/log/syslog, /var/log/messages)

2.5. Configuration d’un service ftp anonyme

La mise en place d’un service ftp anonyme demande plus de manipulations. Vous allez mettre l’environnement dans /home.

1. Vérifiez que le fichier /etc/passwd dispose bien d’un compte ftp :

```
knoppix@master:~/tmp$ grep ftp /etc/passwd
ftp:x:1003:1003:Compte ftp anonyme:/home/ftp:/bin/true
```

sinon modifie les fichier "/etc/passwd" pour ajouter la ligne. Attention, prenez un "UID" libre.

2. Créez un compte de groupe dans le fichier /etc/group :

```
knoppix@master:~/tmp$ grep ftp /etc/group
ftp:x:1003:
```

3. Ulisez la commande "pwconv" pour mettre à jour le fichier shadow.

```
Remarque si vous avez de problèmes d'accès sur le serveur
ftp anonyme par la suite :
Il s'agit peut être d'un problème de définition du compte
dans le fichier "/etc/shadow". Vous pouvez pour cela utiliser
plusieurs options :
A) Première option
1 - taper "pwunconv"
2 - modifier le fichier "/etc/passwd" comme indiqué ci-dessus
3 - taper "pwconv" pour "recacher" les mots de passe.

B) Deuxième option
1 - utiliser la commande "adduser ftp" qui va mettre les fichiers
"/etc/passwd" et "/etc/shadow" à jour. Mettez un mot de passe bidon.
2 - taper "pwunconv" et supprimer le mot de passe du compte dans
"/etc/passwd" (mettre "*" par exemple)
3 - remplacer aussi le shell "/bin/bash" par "/bin/true", enregistrer.
4 - taper "pwconv" pour "recacher" les mots de passe.
5 - supprimer /home/ftp (rm -rf /home/ftp)
6 - continuer la procédure ci-dessous.
```

4. Sous le compte root vous allez créer l'environnement pour le service ftp :

```
cd /home && mkdir -p ftp/lib ftp/bin ftp/pub ftp/incoming ftp/etc
```

On copie le programme ls dans ~ftp/bin. Vous pouvez en mettre d'autres, mais soyez prudent.

```
cp /bin/ls  ftp/bin
```

Il reste à créer les comptes dans un fichier local passwd et group. On y met justes les comptes nécessaires pour l'utilisation des programmes mis dans ~ftp/bin.

```
root@master:/home# grep ftp /etc/group > ftp/etc/group
root@master:/home# grep root /etc/passwd > ftp/etc/passwd
root@master:/home# grep ftp /etc/passwd >> ftp/etc/passwd
```

Vérifiez que vous avez bine les informations dans les fichiers ftp/passwd et ftp/group.

On change les permissions

```
chmod -R 111 ftp/bin ; chmod 111 ftp/etc; chmod 444 ftp/etc/*;\
 chmod 555 ftp/pub; chmod 1733 ftp/incoming
```

On rajoute dans ~ftp/lib les librairies utilisées par les programmes mis dans ~ftp/bin. Vous avez, vous le programme "ls". Les librairies utilisées par le programme ls sont visibles avec la commande "ldd".

Si vous ajoutez d'autres programmes, il faudra y mettre également les bonnes librairies.

```
root@master:/home# ldd /bin/ls
        librt.so.1 => /lib/librt.so.1 (0x4001f000)
        libc.so.6 => /lib/libc.so.6 (0x40031000)
        libpthread.so.0 => /lib/libpthread.so.0 (0x40141000)
        /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Il faut donc copier toutes ces libraires dans ~ftp/lib.

```
cp /lib/librt.so.1 /lib/libc.so.6 /lib/libpthread.so.0 /lib/ld-linux.so.2 ftp/lib
```

Voici donc ce que vous devriez avoir à la fin:

```
root@master:/home# ls -alR ftp
root@mr:/home# ls -alR ftp
ftp:
total 28
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 .
drwxrwsr-x    7 root     root         4096 2003-09-19 12:21 ..
d--x--x--x    2 root     root         4096 2003-09-19 12:22 bin
d--x--x--x    2 root     root         4096 2003-09-19 12:19 etc
drwx-wx-wt    2 root     root         4096 2003-09-19 12:19 incoming
drwxr-sr-x    2 root     root         4096 2003-09-19 12:21 lib
dr-xr-xr-x    2 root     root         4096 2003-09-19 12:19 pub

ftp/bin:
total 76
d--x--x--x    2 root     root         4096 2003-09-19 12:22 .
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 ..
---x--x--x    1 root     root        64428 2003-09-19 12:22 ls

ftp/etc:
total 16
d--x--x--x    2 root     root         4096 2003-09-19 12:19 .
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 ..
-r--r--r--    1 root     root           12 2003-09-19 12:19 group
-r--r--r--    1 root     root           87 2003-09-19 12:19 passwd

ftp/incoming:
total 8
drwx-wx-wt    2 root     root         4096 2003-09-19 12:19 .
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 ..

ftp/lib:
total 1296
drwxr-sr-x    2 root     root         4096 2003-09-19 12:21 .
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 ..
-rwxr-xr-x    1 root     root        82456 2003-09-19 12:22 ld-linux.so.2
-rwxr-xr-x    1 root     root      1104040 2003-09-19 12:22 libc.so.6
-rw-r--r--    1 root     root        81959 2003-09-19 12:22 libpthread.so.0
-rw-r--r--    1 root     root        26592 2003-09-19 12:22 librt.so.1
```

```
ftp/pub:
total 8
dr-xr-xr-x    2 root     root         4096 2003-09-19 12:19 .
drwxr-sr-x    7 root     root         4096 2003-09-19 12:19 ..
```

5. C’est normalement terminé.

2.6. Test de l’accès ftp et sécurisation du service

1. Activez le service dans inetd.conf et lancer le service inetd si ce n’est pas déjà fait.
2. Vérifier que le port est bien ouvert avec la commande netstat

```
root@mr:/home# netstat -atup | grep LISTEN
tcp        0      0 *:ftp                   *:*                     LISTEN      879/inetd
```

3. Vérifier que rien n’interdit l’accès au service ftp dans les fichiers hosts.allow et hosts.deny
4. Commentez le fichier /etc/ftpusers comme ci-dessous :

```
# /etc/ftpusers: list of users disallowed ftp access. See ftpusers(5).
root
#ftp
#anonymous
```

5. Testez et vérifiez le bon fonctionnement de l’accès ftp anonyme (en utilisant le compte "ftp" ou "anonymous" avec la commande :

```
lftp localhost -u anonymous
ou
lftp localhost -u ftp
```

   Si la configuration est correcte, vous devriez avoir comme cela est testé dans la partie qui suit le résultat suivant :

```
root@master:/home# lftp localhost -u ftp
Mot de passe: #Il n’y a pas de mot de passe, faites ENTREE
lftp ftp@localhost:~> ls
total 20
d--x--x--x    2 0        0            4096 May  5 03:35 bin
d--x--x--x    2 0        0            4096 May  5 03:35 etc
drwx-wx-wt    2 0        0            4096 May  5 03:04 incoming
dr-xr-xr-x    2 0        0            4096 May  5 03:32 lib
dr-xr-xr-x    2 0        0            4096 May  5 03:04 pub
```

6. Commentez la ligne anonymous dans le fichier /etc/ftpusers et refaites un essai de connexion.
7. Faites un test en utilisant un compte système existant, par exemple "lftp localhost -u mlx" si mlx est votre compte.
8. Restaurez l’état initial du fichier /etc/ftpusers.

9. Interdisez l'accès ftp avec TCP-Wrapper. Testez l'accès anonyme et en utilisant un compte authentifié.
10. Que peut-on conclure des deux méthodes de protection ?

2.7. telnet, ftp et la sécurité

On désactive en général ces services sauf cas très particulier, car les transactions ne sont pas encryptées. On préfère utiliser les services ssh, scp et sftp. Vous devez avoir un service sshd actif sur le serveur.

Exemple d'utilisation ssh :

```
[root@uranus etc]# grep ssh /etc/services
ssh             22/tcp                          # SSH Remote Login Protocol
ssh             22/udp                          # SSH Remote Login Protocol
ssh -l NomUtilisateur Machine
ssh -l mlx localhost
```

Remarque : Sur la version Live-On-CD, vous devez lancer le démon, car il n'est pas actif par défaut : /etc/init.d/sshd start. Le lancement de ce serveur permet l'utilisation de ssh et de scp à partir de clients.

Exemple d'utilisation de sftp :

```
[root@uranus etc]# grep sftp /etc/services
sftp            115/tcp
sftp            115/udp

[root@uranus etc]# sftp
usage: sftp [-1vC] [-b batchfile] [-osshopt=value] [user@]host[:file [file]]
[root@uranus etc]# sftp  mlx@localhost
Connecting to localhost...
mlx@localhost's password:

Vous pouvez ensuite envoyer ou récupérer des fichiers entre les 2 machines.
```

Exemple d'utilisation de scp :

```
SYNOPSIS
     scp [-pqrvC46] [-S program] [-P port] [-c cipher] [-i identity_file]
         [-o option] [[user@]host1:]file1 [...] [[user@]host2:]file2

# Exporte le fichier "unfichierlocal"
 scp -S ssh unfichierlocal  mlx@hotedistant:/un/chemin/distant/unfichierdistant
# Importe de "hotedistant", le fichier distant "unfichierdistant"
 scp -S ssh mlx@hotedistant:/un/chemin/distant/unfichierdistant unfichierlocal
```

La première ligne exporte un fichier, la deuxième importe. Le compte utilisé est mlx. La transaction est encryptée avec ssh.

Les fi chiers hosts

Mettre en place et comprendre la résolution de nom.

1. Présentation

L’atelier présente la résolution de noms d’hôtes sur un petit réseau à l’aide d’un fichier hosts.

Vous utiliserez la commande "ping" pour diagnostiquer le fonctionnement du réseau.

Il est en 3 parties:

- une présentation de la résolution de nom sur un réseau local
- un TP
- un questionnaire

1.1. Avant de démarrer

Vous devez connaître la classe d’adresse de votre réseau. Vous devez connaître également les adresses des hôtes que vous voulez adresser ainsi que leurs noms d’hôtes.

1.2. Fiche de cours

Dans un réseau, on assigne généralement un nom à chaque hôte. Le terme d’hôte est pris dans son sens large, c’est à dire un "noeud de réseau". Une imprimante, un routeur, un switch, un serveur, un poste de travail sont des noeuds qui peuvent avoir un nom d’hôte, s’ils ont une adresse IP.

On parle de "nom d’hôte" sur les réseaux qui utilisent le protocole TCP/IP. Ne pas confondre, donc, le "nom d’hôte" avec le "nom Netbios" qui est utilisé sur les réseaux Microsoft ou IBM.

Le nom d'hôte est assigné à un noeud qui est configuré avec une adresse IP. Le nom permet d'adresser le noeud, autrement qu'avec l'adresse IP. Par exemple, si le réseau est équipé d'un serveur d'adresse 192.68.0.100 et dont le nom d'hôte est "srv1", il sera alors possible de taper les commandes suivantes:

- telnet 192.68.0.100 ou bien

- telnet srv1

Le nom sert de mnémonique, qui évite de retenir toutes les adresses IP du réseau. Le protocole TCP/IP se charge de la résolution des noms d'hôtes, ensuite le protocole arp, se charge de la résolution des adresses IP en adresses Ethernet.

Pour que la résolution de nom fonctionne, il faut déclarer dans un fichier, tous les noms d'hôtes, et pour chaque nom, son adresse IP. Cette déclaration est réalisée dans le fichier "/etc/hosts".

Remarque: Le processus de résolution équivalent peut être mis en oeuvre sur des réseaux qui utilisent Windows 9x, Windows NT Server, Windows NT Workstation. Vous devrez alors créer les fichiers respectivement dans les répertoires Windows et winnt\system32\drivers\etc. Vous trouverez dans ces répertoires, si TCP/IP est installé un fichier "host.sam" qui peut vous servir d'exemple.

2. TP

Vous utiliserez un éditeur joe ou emacs afin de modifier le fichier /etc/hosts. Utilisez l'algorithme suivant pour créer/modifier votre fichier :

```
Pour chaque hôte du réseau faire
   mettre un enregistrement
Fin pour
```

Les enregistrements ont la structure suivante : AdresseIP Nom1 [...NomN]

Exemple : 195.115.88.35 foo foo.foo.org becassine

Consultez également la commande "man hosts"

- Etablissez la nomenclature des machines du réseau. Configurez le fichier host avec la nomenclature. Testez la résolution de nom avec la commande "ping", puis en utilisant les services "telnet" et "ftp".
- Modifiez la correspondance Nom/Adresse d'une des machines que vous avez dans votre fichier host et accédez y avec telnet. Que se passe-t-il ?
- Débranchez la jarretière de votre carte réseau, et réutilisez les commandes "ping localhost", "ping 127.0.0.1", "ping UneMachineDistante". Que se passe-t-il et que peut-on en déduire ?

Attention, plus tard nous verrons la résolution de nom par un autre service "DNS". Les deux solutions (DNS et Hosts) ne sont pas exclusives, par contre on peut jouer sur l'ordre qui doit être appliqué. Cela est traité par le fichier de configuration "/etc/host.conf".

```
$ more /etc/host.conf
order hosts,bind
```

Il faut bien se souvenir de ça, car dans l'exemple donné ci-dessus, le fichier "hosts" est prioritaire sur "DNS".

3. Questions

- Quelle est la commande qui permet d'obtenir le nom d'hôte de la machine locale ?
- Quelles sont les informations que donne la commande ifconfig ?
- Donnez la commande qui permet de n'envoyer qu'un seul ping à une machine distante (voir man ping)
- Quelle est la taille d'un paquet envoyé par la commande ping ?
- Quelle est la commande qui permet d'envoyer des paquets de 1500 octets ?
- Quelle est la commande ping qui permet d'envoyer des paquets en flot ininterrompu ?
- Quel protocole utilise la commande ping ?

Installation d’un serveur HTTP

Configuration d’un serveur Apache et mise en place de services web

1. Accès aux archives

Vous pourrez récupérer les documents nécessaires sous forme d’archive sur le serveur de linux-france. Pour cela voir la page d’introduction du document.

2. Résumé

Installation et configuration d’un serveur HTTP avec Apache.

Mots clés « Serveur Web », « Serveur HTTP », « Apache »

Description et objectifs de la séquence

Le document doit vous permettre de mettre en place un serveur Apache supportant :

- des accès anonymes,
- des accès authentifiés par Apache,
- un accès à des pages personnelles.
- mettre en place des scripts CGI.
- mettre en place des serveurs web virtuels.

3. Présentation du serveur Apache

Ce chapitre donne un aperçu des fonctions et de l’environnement du serveur Apache. Vous pourrez retrouver tous les aspects développés dans la documentation du produit.

Il existe des outils graphiques de configuration et d’administration d’Apache. Vous allez réaliser les TP(s) de cet atelier en mode commande.

3.1. Présentation de l'environnement

- le binaire apachectl est dans /usr/sbin,
- les fichiers de configuration sont dans /etc/apache/
- la documentation est dans /usr/share/doc
- Le script de lancement du service serveur dans /etc/init.d
- Les journaux sont dans /var/log/apache/

Faites une copie de sauvegarde des fichiers de configuration avant toute manipulation.

3.2. Installation d'un service minimum

Ce paragraphe décrit les principaux paramètres pour mettre en place un service HTTP minimum, avant de lancer le service serveur. Vous utiliserez le fichier de configuration d'Apache httpd.conf.

- port 80, indique quel est le port utilisé par le service (par défaut 80). Il est possible d'utiliser un autre port, par contre vous devrez spécifier au navigateur quel est le port utilisé par le serveur. Si vous configurez par exemple le port 8080 sur une machine www.MonDomaine.edu, vous devrez spécifier dans le navigateur www.MonDomaine.edu:8080, pour que le serveur reçoive et traite votre requête.
- user www-data et group www-data, spécifient le compte anonyme utilisé par le serveur une fois qu'il est lancé. En effet, pour accéder aux ports inférieurs à 1024, le serveur utilise un compte administrateur, ce qui présente des dangers. Une fois le processus actif, il utilisera l'UID d'un autre compte (ici nobody). Ce compte doit pouvoir lire les fichiers de configuration et ceux de la racine du serveur HTTP. D'autres distributions utilisent le compte "nobody" ou "apache"
- ServerAdmin root@localhost, précise quel est le compte qui reçoit les messages. Par défaut le compte administrateur sur la machine locale (à modifier pour une adresse comme root@MonDomaine.edu).
- ServerRoot /etc/apache, indique l'adresse du répertoire racine du serveur, où sont logés les fichiers de configuration du serveur HTTP. Cette adresse peut être modifiée.
- ErrorLog, fichier error_log, journalisation des erreurs. L'adresse est calculée à partir de ServerRoot. Si ServerRoot est /etc/httpd et ErrorLog logs/error_log, le chemin complet est /var/log/apache/logs/error_log.
- ServerName www.MonDomaine.edu, indique le nom ou l'alias avec lequel la machine est désignée. Par exemple, l'hôte ns1.MonDomaine.edu, peut avoir le nom d'alias www.MonDomaine.edu. Voir la résolution de nom avec un DNS.
- DocumentRoot /var/www, indique l'emplacement par défaut des pages HTML quand une requête accède au serveur. (exemple : la requête http://www.MonDomaine.edu/index.html pointe en fait sur le fichier local /home/httpd/html/index.html).
- ScriptAlias /cgi-bin/ /usr/lib/cgi-bin, de la forme "ScriptAlias FakeName RealName", indique où sont physiquement situés les scripts sur le disque, ainsi que l'alias utilisé par les développeurs pour le développement des scripts et des pages. Un développeur utilisera un lien (exemple : /cgi-bin/NomDuScript où /cgi-bin/ est un alias sur /home/httpd/cgi-bin/), et c'est le script /home/httpd/cgi-bin/NomDuScript qui sera

effectivement exécuté. La mise en place d'alias permet de restructurer ou déplacer un serveur sans avoir à modifier toutes les pages développées.

- UserDir public_html, ce paramètre décrit le processus utilisé pour accéder aux pages personnelles d'une personne, si ces pages sont stockées dans son répertoire personnel. Supposons que vous êtes l'utilisateur "bestof" du réseau et que vous ayez des pages personnelles. Il sera possible d'accéder à vos pages, avec l'adresse suivante: www.MonDomaine.edu/~bestof/index.html. Le (tilde "~") permet d'accéder à votre répertoire personnel. La requête sera réellement exécutée sur "/home/bestof/public_html/index.html.

  Attention, vérifier que le répertoire personnel ne soit pas en mode 700, car personne ne pourrait accéder aux pages personnelles.

- Alias /CheminVu/ /CheminRéel/, par exemple : "/icons/ /usr/share/apache/icons/", ce paramètre permet de renommer, à la manière d'un lien logique, un emplacement physique avec un nom logique.

  Exemple: vous voulez que www.MonDomaine.edu/test/index.html, ne corresponde pas physiquement à un répertoire sur la racine du serveur HTTP mais à un emplacement qui serait /usr/local/essai. Vous pouvez mettre dans le fichier de configuration d'Apache un alias de la forme: alias /test/ /usr/local/essai/

- DirectoryIndex donne le ou les noms des fichiers que le serveur doit rechercher si le navigateur passe une requête sur un répertoire. Par exemple sur une requête http://www.MonDomaine.edu, le serveur va rechercher dans l'ordre s'il trouve un fichier index.html, index.stml, index.cgi... en fonction des paramètres de cette variable.
- Les fichiers .htaccess : Apache permet de sécuriser les accès répertoire par répertoire. Il est possible de définir, le nom du fichier qui contiendra les possibilités d'accès par un utilisateur à un répertoire donné. Par défaut la valeur est .htaccess. Ce paramètre est modifiable.
- Limitations de la sécurité par répertoire: Ce procédé alourdit la charge du serveur. En effet, si une requête est passée sur www.MonDomaine.edu/rep1/rep2/index.html, le serveur va vérifier dans chaque répertoire rep1, rep2... l'existence d'un fichier .htaccess. Ce sont les règles du dernier fichier qui seront appliquées. Ce processus est mis en oeuvre pour chaque accès. Cette directive est donc à utiliser avec beaucoup de parcimonie car elle crée une surcharge pour le serveur.

  La directive AllowOverride None, permet de désactiver l'utilisation des fichiers .htaccess dans les niveaux inférieurs. La directive AllowOverride peut être utilisée avec d'autres options par exemple: AuthConfig.

  Les fichiers .htaccess peuvent, s'ils sont présents spécifier leurs propres directives d'authentification,

  La directive ExecCGI, permet l'exécutiuon de script cgi dans ce répertoire.

- Sécuriser un répertoire en autorisant/refusant l'accès

  Pour chaque répertoire "UnRépertoire", sur lequel on désire avoir une action, on utilisera la procédure suivante:

```
<Directory UnRépertoire>
...Ici mettre les actions...
</Directory>
```

Tout ce qui est entre les balises s'applique au répertoire "UnRépertoire".

Exemple: On désire supprimer l'accès du répertoire "/intranet" à tout le monde sauf pour les machines du réseau d'adresse 192.168.1.0 et de nom de domaine MonDomaine.edu.

```
<Directory /intranet>
#Ordre de lecture des règles
order allow,deny
deny from all
allow from 192.168.1 #ou encore allow from .MonDomaine.edu
</Directory>
```

Il importe de préciser dans quel ordre les règles de restriction vont être appliquées. Cet ordre est indiqué par le mot réservé « order », par exemple « order deny,allow » (On refuse puis on alloue l'accès à quelques adresses) ou « order allow,deny » (on accepte généralement les accès mais il sont refusés pour quelques adresses).

Exemple: On désire que l'accès soit majoritairement accepté, sauf pour un ou quelques sites.

```
<directory /home/httpd/html>
AllowOverride none
Order deny,allow
deny from pirate.com badboy.com cochon.com
allow from all
</directory>
```

- Authentifier l'accès à un répertoire : Ce procédé va permettre de sécuriser l'accès à un répertoire ou à des fichiers. L'accès sera autorisé à une ou plusieurs personnes ou encore à un ou plusieurs groupes de personnes.

AuthName, définit ce qui sera affiché au client pour lui demander son nom et son mot de passe,

AuthType, définit le mode d'authentification et d'encryptage « basic » avec HTTP/0 ou « MD5 » par exemple avec HTTP/1.

AuthUserFile, définit le fichier qui contient la liste des utilisateurs et des mots de passe. Ce fichier contient deux champs (Nom d'utilisateur, Mot de passe crypté). Vous pouvez créer ce fichier à partir du fichier /etc/passwd (attention ! faille de sécurité. Il n'est pas forcément avisé d'avoir le même mot de passe pour accéder à Linux et pour accéder à un dossier Web) ou avec la commande "htpasswd" d'Apache.

Exemple du mode d'utilisation de la commande "htpasswd" :

```
root@mr:/home# htpasswd --help
        htpasswd [-cmdps] passwordfile username
 -c  Create a new file.

#> htpasswd -c /etc/apache/users mlx
Ici on crée le fichier /etc/apache/user et on ajoute un compte.
N'utiliser l'option "-c" que al première fois.
```

AuthGroupFile définit le fichier qui contient la liste des groupes et la liste des membres de chaque groupe,

Require, permet de définir quelles personnes, groupes ou listes de groupes ont une permission d'accès.

Exemple de fichier AuthUserFile :

```
doudou:zrag FmlkSsSjhaz
didon:PsddKfdqhgf.fLTER
```

Exemple de fichier AuthGroupFile :

```
users: tintin milou haddock dupond dupont tournesol
tournesol dupont
```

Exemple d'autorisation :

```
require user tintin dupond /* tintin et dupond ont un accès */
require group users /* le groupe users à un accès */
require valid-user /* toute personne existant dans AuthUserFile */
```

Exemple d'accès sécurisé sur un répertoire :

```
<Directory /home/httpd/html/intranet/>
AuthName PatteBlanche
AuthType basic
AuthUserFile /etc/httpd/conf/users
AutGroupFile /etc/httpd/conf/group
     <Limit GET POST>#Ici il faudra un mot de passe
require valid-user
     </Limit>
</Directory>
```

Voici la fenêtre sécurisée que propose Netscape sur l'URL http://localhost/essai:

Figure 10. Accés sécurisé sur un répertoire par Apache

La déclaration d’un accès authentifié sur un répertoire est faite dans le fichier de configuration d’Apache, ou alors en créant un fichier ".htaccess" dans le répertoire que l’on souhaite sécuriser. Le fichier ".htaccess" contient les mêmes directives que celles qui auraient été déclarées dans le fichier httpd.conf.

Attention :

```
Si vous mettez les clauses d’accès restreint pour un
répertoire dans le fichier de configuration d’Apache, les clauses
seront incluses entre 2 balises :
<Directory ...>
</Directory ...>

Si vous mettez les clauses de restrictions dans un fichiers ".htaccess",
vous n’avez pas besoin de mettre ces balises.
```

3.3. Activation du serveur

Utilisez les commandes suivantes pour activer, désactiver le serveur:

/etc/init.d/apache start

/etc/init.d/apache stop

/etc/rc.d/init.d/httpd status

Pour relire le fichier de configuration alors qu’apache est déjà lancé, utilisez :

/etc/init.d/apache reload

Pensez dans tous les cas à consulter les journaux afin de détecter les dysfonctionnements.

3.4. Test de la confi guration

Lancez le navigateur et tapez l’url http://localhost. Vous devriez pouvoir utiliser indifféremment l’adresse IP ou le nom d’hôte de votre machine. Vous devez également pouvoir accéder à partir des autres machines de la salle.

4. Questions

- Quel protocole et quel port utilise le serveur Apache ?
- Comment se nomme le principal fichier de configuration d'Apache, et où se trouve-t-il ?
- Dans quel répertoire sont situées les pages du serveur ?
- Vous modifiez le port d'utilisation du serveur et vous faites un essai à partir d'un client. L'accès ne fonctionne pas. Donnez au moins deux raisons possibles et les moyens de remédier à ce problème.
- Quel est le paramètre qui permet l'utilisation de répertoires personnels pour les utilisateurs afin de déployer leurs sites WEB personnels ?
- Vous activez le paramètre du répertoire personnel dans Apache et relancez le serveur. Vous essayez l'accès sur votre compte or il est refusé. Que se passe-t-il et comment corriger le problème ?
- Dans quel répertoires se trouvent les fichiers log d'Apache et comment se nomment ces fichiers ?

Installation d’un serveur HTTP - TP

1. Résumé

Installation d’un serveur WEB - TP(s)

La séquence est bâtie pour des travaux réalisés avec plusieurs machines. Certaines parties pourront être réalisées sur votre propre machine, celle-ci servant de client et de serveur.

Vous devez avoir un navigateur d’installé, par exemple mozilla, konqueror, galeon...

La résolution de nom doit fonctionner.

Attention : Les paramètres peuvent différer d’une version à l’autre de Linux ou d’une distribution à l’autre. J’utilise dans ce document des variables, vous devrez y substituer les valeurs réelles de votre environnement.

- $APACHE_HOME, répertoire dans lequel sont stockées les pages du serveur.
- $APACHE_CONF, répertoire dans lequel sont stockés les fichiers de configuration.
- $APACHE_USER, compte utilisateur sous lequel fonctionne Apache.
- $APACHE_GROUP, groupe auquel est rattaché le compte $APACHE_USER.

2. TP1 - Installation d’un serveur Web

2.1. Introduction

Vous allez réaliser les opérations suivantes:

- configurez le serveur HTTP pour qu’il soit activé en mode standalone
- activez le serveur HTTP,
- testez le fonctionnement du serveur

A la fin vous devriez pouvoir accéder sur toutes les machines (serveurs HTTP) du réseau à partir du navigateur client.

Attention Vous utiliserez les éléments donnés dans la fiche de cours.

2.2. Configuration du serveur

Vous allez réaliser une configuration de base du serveur. Vous allez donc modifier le fichier httpd.conf. Avant toute modification, faites une copie de sauvegarde des fichiers.

Ouvrez le fichier à l'aide d'un éditeur, relevez et vérifiez les paramètres suivants. Pour chacun de ces paramètres vous noterez leurs rôles à partir des commentaires donnés par les fichiers httpd.conf. (pensez à enregistrer vos modifications):

- ServerName, nom pleinement qualifié de votre serveur
- ServerType standalone
- Port 80
- User $APACHE_USER
- Group $APACHE_GROUP
- ServerAdmin root@localhost
- ServerRoot /etc/apache
- DocumentRoot $APACHE_HOME/html
- UserDir public_html
- DirectoryIndex index.html index.shtml index.cgi
- AccessFileName .htaccess
- Alias /icons/ $APACHE_HOME/icons/
- ScriptAlias /cgi-bin/ $APACHE_HOME/cgi-bin/

2.3. Activation du serveur

Regardez dans la fiche de cours les commandes de lancement du service serveur et la procédure de test. Regardez dans les fichiers de log et dans la table de processus si le service est bien démarré. Vérifier avec la commande netstat que le port 80 est bien ouvert.

Notez toutes les commandes que vous utilisez.

2.4. Test de la configuration

A ce stade le serveur est configuré et fonctionne. Il ne reste plus qu'à réaliser les tests. Vous devez pour cela activer le navigateur.

Faites les tests à partir de la machine locale et d'une machine distante. Utilisez les adresses localhost, 127.0.0.1, les adresses IP et les noms d'hôtes.

Si tout fonctionne vous êtes en mesure de déployer votre site. Il suffira pour cela de l'installer dans l'arborescence $APACHE_HOME.

Dépannage: si cela ne fonctionne pas, procédez par élimination.

- 1 - Essayez avec les adresses IP des machines. Si ça fonctionne c'est que la résolution de nom n'est pas en place.
- 2 - Vérifiez que votre serveur est bien actif.
- 3 - Vérifiez que la configuration du serveur est correcte. Si vous apportez des modifications vous devez réinitialiser le serveur HTTP.

2.5. Auto-évaluation sur le premier TP

- Quels sont le/les fichiers de base pour la configuration du serveur apache et dans quels répertoires sont-ils situés ?
- Comment se nomme le compte d'utilisateur qui utilise le serveur http ?
- Quels sont les permissions d'accès par défaut sur le site principal du serveur ?
- Dans quel répertoire sont installés par défaut les pages HTML du site ?
- Quels sont les deux modes de lancement du serveur ?
- Dans quel fichier détermine-t-on ce mode de fonctionnement ?
- Dans quel répertoire par défaut sont stockés les scripts CGI et quel en est l'alias ?
- Quel est le principal rôle des alias ?
- Quelle(s) procédure(s) peut-on utiliser pour déterminer l'état du serveur et son bon fonctionnement ?
- Vous installez un serveur Apache sur une machine d'adresse 192.168.90.1 et de nom foo.foo.org. Lors des tests sur la machine locale, les commandes http://localhost, http://127.0.0.1, http://192.168.90.1 fonctionnent et http://foo.foo.org ne fonctionne pas. Lors des tests à partir d'une machine distante les commandes http://192.168.90.1 et http://foo.foo.org fonctionnent.

  Que peut-on en déduire et comment résoudre le problème ?

3. TP 2 - Création de pages Web

3.1. Résumé

Vous allez réaliser les opérations suivantes

- Vérifiez que la configuration de votre machine est correcte,
- Développez quelques pages HTML puis les déployer,
- Testez les nouvelles pages à partir d’un client Linux et Windows.

Vous utiliserez les archives qui vous sont fournies. Ces archives sont composées des quelques pages html qui vous serviront de site web et de scripts cgi.

3.2. Vérification de la configuration

Installez le service serveur et vérifiez qu’il est bien configuré et actif.

Pour tester la configuration de votre serveur, vous pouvez également utiliser la procédure suivante à partir de l’hôte local ou d’un hôte distant.

Lancez la commande suivante "$ telnet @IP du PC 80" (exemple : telnet 192.168.1.1 80 si cette adresse est celle de votre machine)

Cette commande crée une connexion au serveur httpd (port 80). Ce dernier invoque un agent.

Identifiez la connexion réseau dans une autre fenêtre xterm et avec la commande :

```
netstat -atup | grep ESTABLISHED
root@mr:/home# netstat -atup | grep ESTABLISHED
#Vous devriez obtenir quelque chose comme :
tcp        0      0 mr:33073   mr:www           ESTABLISHED 1513/telnet
tcp        0      0 mr:www     mr:33073         ESTABLISHED 1508/apache
```

Ensuite, transmettez à l’agent la ligne (commande) suivante : "GET /index.html"

Vérifiez que l’agent transmet de manière transparente le document HTML, et qu’il coupe automatiquement la connexion.

3.3. Installation d’un site Web

Vous allez utiliser les documents HTML fournis en annexe. Vous allez procéder de la façon suivante:

- Créez un répertoire $APACHE_HOME/journal pour y mettre toutes les pages html
- Copiez les images dans $APACHE_HOME/icons (normalement /usr/share/apache/icons)
- Copiez le script cgi compilé "prog" dans $APACHE_HOME/cgi-bin (normalement /usr/lib/cgi-bin)

Testez le site à partir d’un navigateur avec la commande http://@URLDuServeur/journal/

Le site est maintenant déployé, testez l’enchaînement des pages, l’affichage des images.

Le formulaire ne fonctionne pas encore. Vous avez copié le script compilé "prog" dans "/usr/lib/cgi-bin". Vérifiez quel est le nom de script que le formulaire "form.html" essaie de lancer sur le serveur.

Modifiez le nom du script ou le formulaire en conséquence.

Testez le fonctionnement du formulaire dans un navigateur.

Si vous rencontrez des difficultés sur l’exécution du script, vérifiez dans le fichier de configuration d’apache que vous avez bien :

```
    ScriptAlias /cgi-bin/ "/usr/lib/cgi-bin/"
et
    <Directory "/usr/lib/cgi-bin">
        AllowOverride None
        Options None
        Order allow,deny
        Allow from all
    </Directory>
```

3.4. Développement d’un site

Réalisez sous LINUX votre curriculum vitae en langage HTML. Celui-ci devra être composé de plusieurs documents reliés par des liens (ancres). Il sera installé dans $APACHE_HOME/cv et les images dans le répertoire référencé par l’alias /icons/.

- - 1 page d’accueil de présentation avec les liens sur les autres pages,
- - 1 page pour la formation initiale,
- - 1 page pour les expériences professionnelles,
- - 1 page pour les loisirs, passions...

Chaque page doit vous permettre de revenir à la page d'accueil.

Mettez les pages dans le répertoire qui était prévu $APACHE_HOME/cv

3.5. Test de vos pages

Vous allez vous connecter à votre site à partir d'un client distant. Utilisez de préférence les adresses URL. Corrigez les erreurs si l'accès n'est pas réalisé.

3.6. Utilisation des alias

Afin de comprendre le fonctionnement des alias vous allez maintenant réaliser quelques manipulations. Vous allez déplacer le répertoire qui contient les images de "$APACHE_HOME/icons" vers "/tmp/httpd/icons.

- Réalisez l'opération de déplacement du répertoire vers /tmp/httpd/icons,
- Apportez les modifications nécessaires aux fichiers de configuration d'Apache
- Vérifiez le résultat.

Vous constatez ainsi qu'il est possible de déplacer un répertoire sur un serveur, sans qu'il soit nécessaire pour autant de modifier toutes les pages utilisant ce répertoire.

3.7. Auto évaluation sur le deuxième TP

- Quel est le nom de la page par défaut qui est ouverte par le navigateur dans un répertoire du serveur HTTP.
- Quel intérêt procure l'utilisation des alias ?
- Dans quels répertoires sont, par défaut installés les pages HTML, scripts CGI, images et comment se nomment les alias ?
- On crée un répertoire $APACHE_HOME/html/journal pour y stocker des pages HTML. Il n'est pas possible d'y accéder alors que pour les autres sites tout fonctionne. Voici le message renvoyé par le navigateur. Aucune mesure de sécurité n'a été mise en oeuvre.

  ```
  Forbidden
  you don't have permission to access / on this server
  ```

  Quelle est la cause du problème et comment y remédier ?

4. TP3 - Confi guration des répertoires personnels

Vous allez mettre en place un accès pour les utilisateurs du système. Ceux-ci auront la possibilité de mettre leurs pages personnelles dans leurs répertoires privés.

Vous allez réaliser les opérations suivantes:

- Configurez le compte personnel,
- Développez un site personnel,
- Testez l’accès au site personnel.

Relevez dans le fichier de configuration d’Apache le nom du répertoire dans lequel doivent être stockées les pages personnelles.

4.1. Confi gurer le compte personnel

- Créez un compte d’utilisateur. Je vais utiliser, pour la description des opérations le compte "mlx",
- Allez dans le répertoire personnel /home/mlx,
- Créez le répertoire du site Web personnel ,
- Dans ce répertoire vous allez créer un répertoire pour les pages, un pour les images, un pour les scripts CGI avec la commande mkdir.

4.2. Développer un site personnel

Vous allez utiliser les pages HTML fournies en annexes. Utilisez les documents du TP précédent si vous en avez besoin.

Installez les fichiers fournis en annexe dans les répertoires adéquats.

Modifiez les pages HTML à l’aide d’un éditeur pour qu’elles utilisent les images de votre répertoire personnel "~/public_html/images" et le script CGI de votre répertoire personnel "~/public-html/cgi-bin" et pas ceux qui sont dans "$APACHE_HOME/cgi-bin".

Pour autoriser l’utilisation de scripts CGI dans un répertoire, vous devez le déclarer pour le serveur Apache. Voici trois exemples :

```
<Directory /home/*/public_html/cgi-bin>
    Options ExecCGI
    SetHandler cgi-script
```

```
</Directory>
<Directory /home/*/public_html>
                Options +ExecCGI
</Directory>
<Directory /var/www/journal>
                Options +ExecCGI
</Directory>
```

Recherchez égelement la ligne dans le fichier httpd.conf :

```
AddHandler cgi-script .cgi .sh .pl
```

Décommentez là ou ajoutez la si ell en’y est pas.

Copiez le script dans le répertoire que vous réservez pour les scripts. Vérifiez si c’est un programme "C" compilé qu’il porte bien l’extension ".cgi", au besoin renommez le.

4.3. Tester l’accès au site personnel

Vous pouvez maintenant tester votre site personnel. A l’aide d’un navigateur utilisez l’URL "http//localhost/~mlx", (remarquez l’utilisation du "~" pour définir le répertoire personnel.)

Corrigez toutes les erreurs que vous pouvez rencontrer (problèmes d’alias, page principale, page de liens, problème de scripts, permissions d’accès au répertoire...)

Faites le test avec les sites personnels situés sur les autres machines.

4.4. Auto-évaluation sur le troisième TP

- Quel avantage présente l’utilisation des répertoires personnels pour le développement de sites Web ?
- Vous installez votre site personnel et vos pages. Vous tentez de réaliser un test or vous n’arrivez pas à accéder à vos pages. Quels peuvent être les problèmes et comment y remédier ?
- Vous rencontrez un problème de configuration. Vous apportez les corrections dans les fichiers de configuration, or la modification n’est toujours pas prise en compte sur le client. Que se passe-t-il et comment corriger le problème ?
- Comment avez vous fait pour que les scripts personnels soient chargés et exécutés de /home/mlx/public_html/cgi-bin
- Pour l’utilisateur mlx, sur la machine saturne et le domaine toutbet.edu, donnez:
    1. l’adresse URL de son site personnel,
    2. l’emplacement physique de son répertoire personnel sur la machine,
    3. le nom (et chemin complet) du fichier qui est activé quand on accède à son site.

5. TP4 - Mise en place d’un accès sécurisé

Vous allez réaliser les opérations suivantes:

- 1 - Déployez un site d’accès en ligne
- 2 - Sécurisez l’accès à ce site par un mot de passe
- 3 - Testez la configuration.

5.1. Déployer un site d’accès en ligne

Vous allez utiliser les pages fournies en annexe.

Créez un répertoire sur votre machine. "mkdir $APACHE_HOME/protege"

Copiez les pages dans ce répertoire.

5.2. Sécuriser l’accès à ce site par un mot de passe

Dans un premier temps vous allez interdire l’accès à tout le monde. Pour cela vous allez modifier le fichier de configuration d’Apache et y mettre les lignes suivantes :

```
<Directory $APACHE_HOME/protege>
order deny,allow
deny from all
</Directory>
```

Arrêtez puis relancez le serveur et faites un test à partir d’un navigateur. Notez le message qui apparaît. Plus personne n’a accès au site.

Pour mettre un accès sécurisé par mot de passe il manque 2 éléments:

- Modifiez la configuration d’accès au répertoire,
- Créez le mot de passe crypté.

Modifiez la configuration d’accès au répertoire

```
<Directory $APACHE_HOME/protege>
AuthName Protected
AuthType basic
AuthUserFile $APACHE_CONF/users # fichier de mots de passe
<Limit GET POST>
require valid-user   # ici on demande une authentification
</Limit>
</Directory>
```

Créez le mot de passe crypté.

Le mot de passe est un fichier texte à deux champs séparés par un "deux points" (:). Le premier champ contient le compte d’utilisateur, le deuxième contient le mot de passe crypté.

Pour créer ce fichier, les comptes et les mots de passe, utilisez la commande "htpasswd"

5.3. Tester la confi guration.

Ouvrez une session à l’aide d’un navigateur et ouvrez l’URL "http://localhost/protege"

Une fenêtre doit s’ouvrir, entrez le nom d’utilisateur et le mot de passe.

Réalisez les opérations à partir de machines distantes.

Dépannage:

- Vérifiez le nom du répertoire que vous avez créé et la déclaration dans le fichier access.conf,
- Vérifiez le nom et la structure du fichier dans lequel vous avez mis les mots de passe.
- Si vous faites plusieurs tests, quittez puis relancer le navigateur après chaque session ouverte ou refusée,
- Vérifiez que le répertoire soit bien en mode 755 (chmod)
- Si cela ne fonctionne toujours pas reprenez le processus au début:

  - affectez toutes les permissions à tout le monde,

  - supprimez toutes les permissions à tout le monde,

  - affectez les restrictions.

- Utilisez un compte sans mot de passe. Le fichier $APACHE_CONF/users va contenir uniquement la chaîne: mlx, mais il n’y a pas le mot de passe.

Si cela fonctionne alors le problème vient du cryptage du mot de passe.

Une fois que vous avez été authentifié, quittez et relancez le navigateur si vous voulez refaire le test d'accès à la ressource protégée.

5.4. Les fi chiers .htaccess

En vous basant sur ce que vous venez de faire, sécurisez l'accès à un autre répertoire en utilisant un fichier .htaccess.

5.5. Auto-évaluation sur le quatrième TP

- Dans quel cas un accès sécurisé peut-il être utilisé ?
- Vous affectez un mot de passe à un utilisateur distant. Vous faites un test sur votre machine tout semble fonctionner. Lui vous appelle pour vous dire que ses requêtes sur le site protégé sont toujours rejetées. Que se passe-t-il ?
- Si on utilise les possibilités du système, quelle autre solution peut-on utiliser pour interdire l'accès à un répertoire.

6. TP5 - Utilisation de scripts CGI

Il existe deux méthodes qui permettent de faire communiquer un formulaire html avec un script CGI. La méthode POST et la méthode GET. Nous utiliserons ici la méthode POST.

Ce TP doit vous permettre de développer un formulaire et un script CGI en C. Vous devez savoir compiler un programme.

Vous allez réaliser les opérations suivantes:

- Étudiez les sources fournies en annexe.
- Développez un formulaire et adaptez les scripts,
- Testez le fonctionnement des scripts.

6.1. Etudier les sources fournies en annexe

Transférez les programmes C, les .h et le makefile dans votre répertoire personnel. Etudiez attentivement les sources. Testez le fonctionnement du script.

6.2. Développer un formulaire et adapter les scripts

A partir de l'exemple fourni, développez les pages HTML d'un site commercial qui doit permettre la prise de commande à distance de pizzas. Il doit y avoir au moins 3 pages:

- une page principale,
- une page de description des produits,
- une page (formulaire) pour passer commande contenant tous les types de champs qui existent dans le formulaire form.html. (liste déroulante, bouton radio, zone de texte)

Exemple :

- Zone de texte (Nom du client)
- Liste déroulante (Calzone, Margarita, Quatre-saisons)
- Bouton radio (Grande, Moyenne, Petite)
- Boîte à cocher (Avec des anchoix)
- Vous afficherez au client le résultat de sa commande.

Adaptez le script CGI qui vous est fourni, à votre formulaire.

6.3. Tester le fonctionnement de votre script.

Pour tester le script:

- ouvrez la page principale de votre site à l'aide d'un navigateur,
- passez des commandes,
- vérifiez les résultats retournés par le script. (réponse).

6.4. Auto-évaluation sur le cinquième TP

- Que signifie CGI
- Quel intérêt présente l'utilisation de scripts CGI ?
- Quelle est la différence entre la méthode GET et POST ?
- Comment se nomment les variables d'environnement qui contiennent la chaîne (paramètres) du formulaire ?
- Comment est structurée cette chaîne ?
- Quel est le caractère de concaténation des chaînes ?
- Pourquoi la réponse contient dans l'entête la chaîne Content-type: text/html ?
- A quoi correspondent ces 2 paramètres text et html ?

7. TP6 - Serveurs webs virtuels et redirection

Il est préférable d'avoir réalisé les TP sur les serveurs de noms avant de réaliser celui-ci.

Pour réaliser ce TP, vous devrez avoir un serveur de nom qui fonctionne.

La mise en place de serveurs webs virtuels, permet de faire cohabiter plusieurs serveurs sur un même hôte. Nous verrons qu'il y a plusieurs techniques pour faire cela :

- Les serveurs virtuels basés sur le nom, dans ce cas, vous devrez désigner pour une adresse IP sur la machine (et si possible le port), quel est le nom utilisé (directive ServerName), et quel est est la racine du site (directive DocumentRoot).

```
# Ici toutes les adresses ip sont utilisées, on peut mettre *
    NameVirtualHost *

    <VirtualHost *>
    ServerName www.domain.tld
    DocumentRoot /www/domain
    </VirtualHost>

    <VirtualHost *>
    ServerName www.otherdomain.tld
    DocumentRoot /www/otherdomain
    </VirtualHost>

# Ici on utilise une adresse en particulier
# Toutes les requêtes sur http://www.domain.tld/domain
# pointeront sur /web/domain
    NameVirtualHost 111.22.33.44

    <VirtualHost 111.22.33.44>
    ServerName www.domain.tld
    ServerPath /domain
```

```
DocumentRoot /web/domain
</VirtualHost>
```

- Les serveurs virtuels basés sur des adresses ip. Dans ce cas, chaque serveur aura sa propre adresse IP. Vous devrez également avoir un serveur de nom sur lequel toutes les zones et serveurs sont déclarés, car c’est ce dernier qui assurera la correspondance "adresse ip <-> nom du serveur" :

```
# Chaque serveur peut avoir son propre administrateur, ses
# propres fichiers de logs.
# Tous fonctionnent avec la même instance d’Apache.
# Il est possible de lancer plusieurs instances d’apache
# mais sur des ports différents

    <VirtualHost www.smallco.com>
    ServerAdmin webmaster@mail.smallco.com
    DocumentRoot /groups/smallco/www
    ServerName www.smallco.com
    ErrorLog /groups/smallco/logs/error_log
    TransferLog /groups/smallco/logs/access_log
    </VirtualHost>

    <VirtualHost www.baygroup.org>
    ServerAdmin webmaster@mail.baygroup.org
    DocumentRoot /groups/baygroup/www
    ServerName www.baygroup.org
    ErrorLog /groups/baygroup/logs/error_log
    TransferLog /groups/baygroup/logs/access_log
    </VirtualHost>
```

La redirection est un service un peu particulier, qui diffère de celui des services web virtuels. Il permet de rediriger une partie du site web à un autre endroit du disque physique. La encore on utilisera une des fonctions d’Apache qui permet de définir des alias.

Prenons par exemple un site installé physiquement sur "/var/www" et accessible par l’url "http://FQDN". L’url "http://FQDN/unREP" devrait correspondre normalement à l’emplacement physique "/var/www/unREP". La redirection va permettre de rediriger physiquement vers un autre répertoire.

Cette technique est largement utilisée par des applications ou pour la création d’espaces documentaires.

l’URL "http://FQDN/compta", pourra pointer physiquement dans "/usr/lib/compta" au lieu de "/var/www/compta".

Dans cet atelier, vous allez réaliser les opérations suivantes:

1. Installer un service de serveurs web virtuels par adresse et par nom,
2. Mettre en place un service de redirection par alias.

7.1. Avant de commencer sur les serveurs web virtuels

Configurez votre serveur de nom si ce n’est pas fait. Vous pouvez vous aider de l’annexe sur le "web-hosting" ci-dessous.

Pour créer des alias dynamiquement à votre interface réseau, utilisez la commande suivante:

```
ifconfig eth0:0 192.168.0.1
ifconfig eth0:1 192.168.1.1
ifconfig
eth0      Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.90.2  Bcast:192.168.90.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:95041 errors:0 dropped:0 overruns:0 frame:0
          TX packets:106227 errors:0 dropped:0 overruns:26 carrier:0
          collisions:0 lg file transmission:100
          RX bytes:57490902 (54.8 MiB)  TX bytes:11496187 (10.9 MiB)
          Interruption:11 Adresse de base:0x3000

eth0:0    Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.0.1  Bcast:192.168.0.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interruption:11 Adresse de base:0x3000

eth0:1    Lien encap:Ethernet  HWaddr 00:D0:59:82:2B:86
          inet adr:192.168.1.1  Bcast:192.168.1.255  Masque:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          Interruption:11 Adresse de base:0x3000

lo        Lien encap:Boucle locale
          inet adr:127.0.0.1  Masque:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 lg file transmission:0
          RX bytes:4564 (4.4 KiB)  TX bytes:4564 (4.4 KiB)
```

Créez votre serveur de nom pour deux domaines par exemples (domain1.org et domain2.org), adaptez la configuration du fichier "/etc/resolv.conf", vérifiez le bon fonctionnement de la résolution de nom sur les CNAME (www.domain1.org et www.domain2.org).

7.2. Serveur web virtuel basé sur les adresses ip

D’abord les déclarations dans le fichier /etc/apache/Vhosts:

```
# Web hosting ip based
NameVirtualHost 192.168.0.1
<VirtualHost 192.168.0.1>
```

```
DocumentRoot /home/web/domain1
ServerName www.domain1.org
</VirtualHost>

NameVirtualHost 192.168.1.1
<VirtualHost 192.168.1.1>
DocumentRoot /home/web/domain2
ServerName www.domain2.org
</VirtualHost>
```

Inclusion à la fin du fichier de configuration d'apache :

```
Include Vhosts
```

Préparation des sites webs sans trop se casser la tête :

```
# mkdir -p /home/web/domain1 /home/web/domain2
# echo "<H1>Salut domain1 </H1>" > /home/web/domain1/index.html
# echo "<H1>Salut domain2 </H1>" > /home/web/domain2/index.html
```

On relance Apache :

```
/etc/init.d/apache restart
```

C'est terminé, les requêtes sur "http://ww.domain1.org", "http://www.domain2.org", doivent fonctionner et vous renvoyer la bonne page.

7.3. Serveur Web virtuel basé sur le nom

On va utiliser l'adresse ip de ns1.domain1.org pour les URI w3.domain1.org et wiki.domain1.org. w3 et wiki sont deux zones de déploiement de site web différentes, sur un même serveur HTTP.

Modification du fichier Vhosts :

```
# Web hosting name based (basé sur le nom)
#Site secondaire de domain1
 <VirtualHost 192.168.0.1>
ServerName w3.domain1.org
DocumentRoot /home/web/w3
</VirtualHost>
```

```
 <VirtualHost 192.168.0.1>
ServerName wiki.domain1.org
DocumentRoot /home/web/wiki
</VirtualHost>
```

Relancer Apache.

Création des sites web :

```
# mkdir -p /home/web/wiki /home/web/w3
# echo "<H1>Salut w3 </H1>" > /home/web/w3/index.html
# echo "<H1>Salut wiki </H1>" > /home/web/wiki/index.html
```

C’est terminé, les requêtes sur :

```
http://w3.domain1.org
http://wiki.domain1.org
```

doivent retourner les bonnes pages.

7.4. Application sur la redirection

Nous allons créer un espace documentaire pour le domain "domain1.org". Il sera accessible par l’url "http://www.domain1.org/mesdocs", mais il sera localisé dans "/etc/domain1doc".

Créez un répertoire "domain1doc" dans /etc pour ce nouveau projet et mettez y le fichier "apache.conf" ci-dessous:

```
# Création du répertoire
mkdir /etc/domain1doc

# Création du fichier /etc/domain1doc/apache.conf
Alias /mesdocs /etc/domain1doc/
<Directory /etc/domain1doc>
Options FollowSymLinks
AllowOverride None
order allow,deny
allow from all
</Directory>
```

Faites une inclusion de ce fichier dans le fichier de configuration d’Apache et relancez le service Apache.

C’est terminé, toutes les requêtes sur "http://www.domain1.org/mesdocs", ouvriront les pages qui sont dans "/etc/domain1doc".

7.5. Annexe pour le "web-hosting"

```
=== A rajouter dans /etc/bind/named.conf

zone "domain1.org" {
        type master;
        file "/etc/bind/domain1.org.hosts";
        };

zone "domain2.org" {
        type master;
        file "/etc/bind/domain2.org.hosts";
        };



=== fichiers de zone
root@freeduc-sup:/etc/bind# more domain1.org.hosts
$ttl 38400
@                       IN      SOA     domain1.org. hostmaster.domain1.org. (
                        2004052604
                        10800
                        3600
                        604800
                        38400 )
@                         IN      NS      ns1.domain1.org.
ns1                       IN      A       192.168.0.1
www                       IN CNAME ns1
wiki                      IN CNAME ns1
w3                        IN CNAME ns1

root@freeduc-sup:/etc/bind# more domain2.org.hosts
$ttl 38400
@                       IN      SOA     domain2.org. hostmaster.domain2.org. (
                        2004052604
                        10800
                        3600
                        604800
                        38400 )
@                         IN      NS      ns1.domain2.org.
ns1                       IN      A       192.168.1.1
www                       IN CNAME ns1
```

Installation d’un serveur SAMBA

Partage de ressources sous GNU/Linux pour les clients GNU/Linux ou Windows avec le protocole SMB.

1. Résumé

Avec Samba vous allez mettre en place un service de partage de disque pour des clients réseau. Ceux-ci peuvent être sous Linux ou sous Windows. Nous verrons surtout la configuration du service serveur sous Linux, et la configuration des clients sous Windows.

Samba est un produit assez populaire. Il existe de plus en plus d’outils de configuration en environnement graphique qui simplifient les tâches sur un serveur en exploitation (partage de ressources, création de comptes utilisateurs). Comme nous n’en sommes pas là, nous allons réaliser les opérations manuellement.

Vous devez savoir ce qu’est un domaine Microsoft, un groupe de travail, comment fonctionne la résolution de nom NetBIOS avec le protocole NetBIOS, ce qu’est un serveur WINS, un serveur d’authentification (contrôleur de domaine).

2. Eléments d’installation et de confi guration de SAMBA

2.1. Environnement de samba

Samba est installé sur la Freeduc-sup. Si vous n’utilisez pas cette distribution, installez les paquets. Il ne devrait normalement pas y avoir de problèmes de dépendances.

Le paquet installe principalement samba et samba-common :

- le programme nmdb qui assure la résolution de nom NetBIOS et smdb qui assure le partage de ressource SMB/CIFS dans /usr/sbin,
- le script d’initialisation dans /etc/init.d,
- un fichier de configuration /etc/samba/smb.conf,
- une documentation complète dans /usr/share/doc.
- le service de journalisation (log) dans /var/log/samba

- des outils comme smbpasswd pour la création des comptes samba et nmblookup pour vérifier le fonctionnement de la résolution de noms NetBIOS

La commande "dpkg-reconfigure samba", vous demande si samba doit être lancé en mode autonome, choisissez "oui", si un fichier "/etc/samba/smbpasswd" doit être créé, choisissez également "oui". La dernière option vous permet d’avoir une base de données de compte créée automatiquement à partir de la base de compte du fichier "/etc/passwd".

Faites tout de suite une sauvegarde du fichier "/etc/smb.conf".

Dans la suite du document, vous remplacerez $NOM_HÔTE par le nom d’hôte de votre machine qui servira également de nom NetBIOS.

2.2. Le fichier de configuration sous Linux

Voici le fichier de configuration qui nous servira de base de travail. Il va permettre de:

- définir $NOM_HÔTE comme contrôleur de domaine,
- mettre en place l’authentification des utilisateurs,
- partager des disques et une imprimante pour un client Windows,
- partager du "home directory" d’un utilisateur sous Linux comme étant son répertoire personnel sous Windows.

Le fichier de configuration comprend essentiellement deux parties :

- une partie "générale" qui définit le comportement général du serveur et la stratégie adoptée pour les services communs (CPD, mode d’authentification, service WINS),
- une partie "share", qui définit les ressources partagées et les permissions d’accès.

2.3. Les étapes de la configuration du serveur

Nous allons réaliser les opérations suivantes :

- Vérifier et valider le fichier de configuration,
- Déclarer les ressources partagées,
- Créer un compte d’utilisateur pour samba.

Il n’y aura plus qu’à tester la configuration à partir d’un client.

Attention, un compte système n’est pas un compte samba. Faites bien la distinction entre les deux.

2.4. Première étape - Installer le fichier de configuration

Configurer l’environnement de samba par le fichier "/etc/samba/smb.conf" et activez le service avec la commande "/etc/init.d/samba start". Cette opération doit être réalisée chaque fois que le fichier de configuration est modifié. Vérifiez la configuration à l’aide de la commande « testparm | more ».

Corrigez les erreurs éventuelles de configuration.

2.5. Deuxième étape - Déclarer les ressources partagées

Cette opération est réalisée dans la partie "share" du fichier smb.conf. Chaque fois que vous ajoutez ou modifiez une ressource relancez le service serveur.

Pour l’authentification sur un domaine, vous devez créer un répertoire "netlogon", par exemple :

```
mkdir -p /home/samba/netlogon
```

et déclarer la ressource dans la section "share" du fichier "/etc/smb.conf".

2.6. Troisième étape - Créer un compte d’utilisateur autorisé

La création d’un compte d’utilisateur Samba et de son mot de passe est réalisée à l’aide de la commande smbpasswd.

```
smbpasswd -a MonCompte MonMotDePasse
```

ajoute le compte samba "MonCompte" avec le mot de passe "MonMotDePasse" dans le fichier /etc/samba/smbpasswd.

Voir "man smbpasswd" ou "smbpasswd --help" pour le mode d’utilisation de la commande. Le compte système doit être préalablement créé avec la commande unix "adduser".

Trois remarques :

- Les manipulations peuvent paraître fastidieuses si vous avez un grand nombre de comptes utilisateurs.
- Si vous disposez de nombreux comptes d’utilisateurs sur votre système Linux, vous disposez d’un script qui permet de créer automatiquement les comptes Samba à partir du fichier /etc/passwd. Voir "man mksmbpasswd" pour le mode d’utilisation de la commande. Il est également possible de créer sans difficulté un script qui, a partir d’un fichier texte crée les comptes systèmes et les comptes samba.

Toutes les indications sont dans la documentation de Samba.

2.7. La configuration d’un client Windows

La configuration du client Windows ne doit pas poser de difficulté.

Configurez le client pour les réseaux Microsoft afin que l’utilisateur soit authentifié par le serveur de votre domaine. Ici le contrôleur de domaine est le serveur samba. Vérifier la configuration du protocole TCP/IP, relancez la machine. Vous pourrez vous authentifier sur le serveur Samba.

Toutes les commandes "net use, net share...", ou les outils comme le voisinage réseau vous permettent d’accéder aux ressources du serveur. (disques partagés, imprimantes, disque personnels).

Un problème à éviter :

Le compte utilisateur Samba dispose de moins de privilèges que le compte root. Si vous partagez un disque et que vous faites les manipulations sous le compte root, faites attention aux droits, car si "root" est propriétaire (chmod 700), le client samba ne pourra pas accéder au disque.

3. Annexe : Exemple de fichier de configuration de Samba :

```
Annexe 1 : Exemple de fichier de configuration de Samba :
Le fichier ci-dessous est simplifié, vous trouverez de nombreuses autres options
dans la documentation.
================================================================================
# Date: 2003/10/03 12:48:57

# Global parameters
[global]
workgroup = INFOGESTION
netbios name = main_serveur
server string = %h server (Samba %v)
encrypt passwords = Yes
passwd program = /usr/bin/passwd %u
passwd chat = *Enter\snew\sUNIX\spassword:* %n\n *Retype\snew\sUNIX\spassword:* %n\n .
syslog = 0
max log size = 1000
socket options = IPTOS_LOWDELAY TCP_NODELAY SO_SNDBUF=4096 SO_RCVBUF=4096
logon script = logon.cmd
logon path = \\%N\profiles\%u
logon drive = h:
logon home = \\homeserver\%u
domain logons = Yes
os level = 33
preferred master = True
```

```
domain master = True
dns proxy = No
wins server = localhost
invalid users = root

[netlogon]
path = /home/samba/netlogon
read only = yes
        browsable = no


[homes]
comment = Home Directories
read only = No
create mask = 0700
directory mask = 0700
browseable = No

[partage]
comment = Ressource partagbéesA
path = /tmp
read only = No
create mask = 0700
printable = Yes
browseable = No

[tmp]
comment = Partage
path = /tmp
create mask = 0700
directory mask = 0700
guest ok = Yes

[printers]
comment = All Printers
path = /tmp
create mask = 0700
printable = Yes
browseable = No


==========================================================================
```

Installation d’un serveur SAMBA

Partage de ressources sous GNU/Linux pour les clients GNU/Linux ou windows avec le protocole SMB.

1. TP

1.1. Résumé

Ce document décrit comment utiliser un serveur samba comme serveur d’identification et d’authentification pour des clients Windows. Le serveur simule un contrôleur de domaine NT4 Server ou 2000.

Vous utiliserez 2 postes en réseau. Le premier est sous Linux, le second sous Windows. On désire installer et configurer le service de partage de fichiers Samba sous Linux. Le client Windows doit permettre l’identification des utilisateurs sur le serveur en utilisant les mots de passe cryptés.

Cet atelier permet la mise en oeuvre du protocole SMB. Il permet également d’envisager la mise en place du partage de fichiers et d’imprimantes.

1.2. Déroulement des opérations

- Les opérations vont se dérouler en 6 étapes :
- Configuration du fichier /etc/smb.conf et démarrage des services,
- Création d’un compte utilisateur,
- Création du fichier d’authentification pour Samba /etc/smbpasswd,
- Création de ressources disques partagées en lecture et en lecture/écriture,
- Configuration du client Windows 9x,
- Procédure de test.

1.3. Configuration du fichier smb.conf et démarrage des services

Le fichier de configuration smb.conf est dans le répertoire /etc/samba. Faites une copie de sauvegarde de ce fichier puis ouvrez l’original avec un éditeur. Modifiez-le afin que les utilisateurs puissent accéder au répertoire /tmp du serveur en « rw » et à leur répertoire personnel en « rw » également.

Vous utiliserez et adapterez l’exemple donné dans la fiche de cours.

1.4. Création d’un compte utilisateur

Vous allez tout d’abord :

- créer le compte système
- créer le compte samba.

Créez 1 compte système à l’aide de la commande « adduser ».

Pour ce compte système vous créerez un compte samba à l’aide de la commande « smbpasswd ».

1.5. Vérification de la configuration sur le serveur Samba

Démarrez le service. Vous pouvez utiliser la commande "testparm" pour valider la configuration du serveur. Vérifiez également la table des processus et les traces dans le fichier log.

Le fichier "DIAGNOSIS.txt" de la documentation de samba, donne une procédure en 10 points pour vérifier que tout fonctionne. Localisez ce fichier, (en général dans /usr/doc ou dans /usr/share/doc/samba) ouvrez-le avec un éditeur et réalisez la procédure de test qui y est décrite.

1.6. Procédure de test à partir d’un client Linux

Si le serveur foncgtionne correctement et que vous utilisez une FREEDUC-SUP, vous pouvez utiliser le "plug-in" "smb" directement à partir de conqueror.

Lancez konqueror à partir d’un autre client linux et utilisez les commandes "smb://@IP_Du_Serveur/" ou "smb://@IP_Du_Serveur/un_Compte_Samba". Vous devriez avoir une fenêtre équivalente à celle donnée ci-dessous.

Figure 11. Accés à un serveur Samba à partir d’un client Linux

1.7. Procédure de test à partir d’un client windows

La procédure de tests sera réalisée à partir d’un client w98. Pour d’autres types de clients, il sera peut être nécessaire de créer des comptes d’ordinateurs, ou adapter la configuration du serveur Samba. Il faudra se référer à la documentation située en général dans "/usr/share/doc/samba".

Configurez votre client Windows pour qu’il puisse faire partie de votre domaine NT (Panneau de configuration, icône Réseau) déclaré sur votre serveur Linux.

Au démarrage du PC, vous devez avoir, sur le client, une fenêtre qui vous demande de vous identifier dans le domaine défini. Vérifiez l’accès.

Une fois la session ouverte vous devez pouvoir utiliser les outils et commandes suivantes :

- Explorateur,

- Voisinage réseau,

- Démarrer, Exécuter, \\NomDuServeurSamba

- Consultez sur le serveur les fichiers: /var/log/samba/log.%m

- Vérifiez les accès en lecture/écriture sur les espaces disques partagés.

Modification de l'environnement serveur

Créez sur le serveur les espaces supplémentaires /mnt/apps et /mnt/partage. Le premier est en lecture uniquement, le deuxième en lecture/écriture. Modifiez smb.conf, relancez le service serveur, testez les accès.

1.8. Automatisation de création de compte.

On donne, dans un fichier texte "personnes", une liste de personnes. Le fichier a la structure suivante :

```
Nom Prénom
```

par exemple :

```
TUX Junior
TUX Tuxinette
TUX Padre
...
```

En général un fichier d'importation n'est pas aussi simple car on peut avoir des prénoms composés, ou des noms comprenant des "espaces". Les champs sont distingués par des séparateurs comme un point-virgule par exemple. Il faudra dans ce cas traiter différement le fichier.

Le principe est simple :

```
- On crée un script qui va créer automatiquement les comptes systèmes,
- Le compte est représenté par la concaténation du nom et du prénom
- Le mot de passe par défaut est la concaténation du nom et du prénom
- Les groupes sont préalablement créés.
```

Vous pouvez avoir des différences entre les distributions type RedHat ... ou Debian. La commande "passwd" par exemple ne supporte pas l'option "--stdin". Vous avez donc ci-dessous deux esmples d'applications qui tiennent compte de ces différences. On donne le script suivant qui crée les comptes systèmes :

```
cat persons | while true ; do
#Si la ligne est vide on quitte, il ne faut donc pas de ligne vide dans le fichier
read ligne
        if [ "$ligne" == "" ] ; then
        exit 0
        fi

#On récupère les paramètes prénom, nom
set -- $ligne
        echo $2 $1
useradd $1$2 -G $1$2
echo $1$2 | (passwd --stdin $1$2)
```

```
done
```

Testez et vérifiez le fonctionnement du script. Modifiez le script pour qu'il crée également les comptes Samba.

Version pour Debian

```
cat persons | while true ; do
read ligne
        if [ "$ligne" == "" ] ; then
        exit 0
        fi
set -- $ligne
        echo $2 $1
addgroup $1$2
useradd  $1$2 -G $1$2
echo $1$2:$1$2 | chpasswd
done
```

1.9. Administration graphique

Vous pouvez utiliser des outils graphiques d'administration de Samba comme swat ou webmin. Pour utilisez swat, décommentez la ligne dans "/etc/inetd.conf" :

```
swat   stream  tcp  nowait.400 root /usr/sbin/tcpd  /usr/sbin/swat
```

Activez les serv ice apache et inetd. Ouvrez le navigateur et tapez :

```
http://localhost:901
```

Ouvrez une session avec le compte root.

Pour utiliser webmin, activez les services apache et webmin. Dans un navigateur allez sur l'url :

```
https://localhost:10000
```

Ouvrez une session avec le compte root et comme mot de passe "knoppix" qui a été mis par défaut sur la Freeduc-Sup. Dans l'onglet "Serveurs", vous pourrez administrer votre serveur samba.

Installation d’un serveur DHCP

Installation et configuration d’un serveur DHCP.

1. Présentation

L’atelier propose

- d’installer un serveur DHCP sous Linux,
- d’installer un client DHCP sous Linux
- d’installer un client DHCP sous Windows
- de réaliser une phase de test avec les commandes winipcfg et ipconfig de Windows

Les éléments sur l’analyse de trame, notamment les trames bootp, seront retraités lors des TP sur la métrologie.

Il est en 4 parties:

1. une présentation du service DHCP
2. l’installation du serveur
3. l’installation des clients
4. les tests

Matériel nécessaire:

Deux machines en dual boot Linux/Windows en réseau.

2. Rôle d’un service DHCP

Un serveur DHCP (Dynamic Host Configuration Protocol) a pour rôle de distribuer des adresses IP à des clients pour une durée déterminée.

Au lieu d'affecter manuellement à chaque hôte une adresse statique, ainsi que tous les paramètres tels que (serveur de nom, passerelle par défaut, nom du réseau), un serveur DHCP alloue à un client, un bail d'accès au réseau, pour une durée déterminée (durée du bail). Le serveur passe en paramètres au client toutes les informations dont il a besoin.

La distribution des adresses par le serveur DHCP aux clients sous la forme de paramètres, montre bien que, tous les noeuds critiques du réseau (serveur de nom primaire et secondaire, passerelle par défaut) doivent avoir une adresse IP statique. Si celle-ci variait, le processus, dans l'état, ne serait pas réalisable.

Ce processus est mis en oeuvre quand vous ouvrez une session chez un fournisseur d'accès Internet par modem. Le fournisseur d'accès, vous alloue une adresse IP de son réseau le temps de la liaison. Cette adresse est libérée, donc de nouveau disponible, lors de la fermeture de la session.

Cela présente plusieurs avantages:

- il n'y a pas à gérer poste par poste les adresses dans le réseau. Chaque noeud vient chercher une adresse quand il en a besoin. Il libère cette adresse quand la session se termine.

- si un paramètre change au niveau du réseau, comme, par exemple l'adresse de la passerelle par défaut, il suffit de changer la valeur du paramètre au niveau du serveur DHCP, pour que toutes les stations aient une prise en compte du nouveau paramètre dès que le bail sera renouvelé. Dans le cas de l'adressage statique, il faudrait manuellement reconfigurer toutes les machines.

- cela permet, dans certains cas de pouvoir adresser plus de postes qu'il n'y a d'adresses IP disponibles. Imaginons un fournisseur d'accès qui a plus de 1000 clients. Il lui faudrait 5 réseaux de classe C, s'il voulait donner à chaque client une adresse IP particulière. S'il se dit que chaque client utilise en moyenne un temps de connexion de 10 mn par jour, il peut s'en sortir avec une seule classe C, en attribuant, ce que l'on pourrait appeler des jetons d'accès en fonction des besoins des clients.

L'inconvénient:

Le client utilise des trames de broadcast pour rechercher un serveur DHCP sur le réseau, cela charge le réseau. Si vous avez une entreprise avec plusieurs centaines de personnes qui ouvrent leur session le matin à 8h ou l'après midi à 14 h, il peut s'en suivre de graves goulets d'étranglement sur le réseau. L'administrateur devra donc réfléchir sérieusement à l'organisation de son réseau.

Ici il faut poser une question:

Comment un client DHCP, qui utilise le protocole TCP/IP mais qui n'a pas encore obtenu d'adresse IP par le serveur, peut-il communiquer sur le réseau ?

La réponse sera abordée pendant les TP sur la métrologie

Vous trouverez des éléments très précis sur le protocole DHCP dans les pages du manuel de Linux. (dhcp3d, dhcpd.conf et dhclient.conf.

3. Indication pour la réalisation du TP

Le processus va se dérouler en 3 étapes:

- la configuration du serveur
- la configuration des clients
- le test de la configuration.

3.1. Installation du serveur

Installation du serveur

Les paquets sont déjà installés.

Attention : Vous pouvez avoir sur votre distribution, plusieurs serveurs DHCP.

dhcpxd, est conforme à la RFC 2131. Il fournit un exemple de configuration assez détaillé.

dhcp3, intègre l'inscription auprès d'un DNS Dynamique. C'est ce package que nous allons utiliser dans le TP. Par contre si vous n'avez pas de DNS dynamique sur le réseau, vous devrez mettre en entête du fichier dhcpd.conf, la ligne :

```
ddns-update-style none;
```

3.2. Configuration du serveur

La configuration consiste à créer 2 fichiers:

- */etc/dhcp3/dhcpd.conf* , ce fichier sert à la configuration même du serveur (plage d'adresses, paramètres distribués)

- */var/lib/dhcp3/dhcpd.leases*, ce fichier va servir à l'inscription des clients. Chaque client DHCP, génère l'écriture d'un enregistrement dans ce fichier. Cela permet le suivi, les statistiques de l'activité du serveur.

3.2.1. Le fichier de configuration dhcpd.conf

Je n'aborde pas tous les paramètres. Je ne donne uniquement qu'un exemple de fichier commenté qui permet de réaliser cet atelier. Vous pouvez créer ce fichier avec un éditeur.

```
$>more dhcpd.conf

# ici il s’agit du réseau 192.168.0.0
subnet 192.168.0.0 netmask 255.255.255.0{

#La plage d’adresse disponible pour les clients
range 192.168.0.10 192.168.0.20;

# Les clients auront cette adresse comme passerelle par défaut
option routers    192.168.0.254;

# Ici c’est le serveur de nom, on peut en mettre plusieurs
option domain-name-servers   192.168.0.1;

# Enfin on leur donne le nom du domaine
  option domain-name   "freeduc-sup.org";

# Et l’adresse utilisée pour la diffusion
option broadcast-address 192.168.0.255;

#Le bail à une durée de 86400 s par défaut, soit 24 h
# On peut configurer les clients pour qu’ils puissent demander
# une durée de bail spécifique
default-lease-time  86400;

#On le laisse avec un maximum de 7 jours
max-lease-time 604800;

#Ici on désire réserver des adresses à des machines
group {

#use-host-decl-names indique que toutes les machines dans l’instruction « group »
# auront comme nom, celui déclaré dans l’instruction host.
use-host-decl-names true ;

# ici définir les machines
host m1 {
hardware ethernet 00:80:23:a8:a7:24;
fixed-address 192.168.0.125;
}
host m2 {
hardware ethernet a0:81:24:a8:e8:3b;
fixed-address 192.168.0.126;
}
    }#End Group
}#End dhcp.conf
```

3.2.2. Création d’un fichier d’inscription

Ce fichier doit parfois être créé, sans quoi le serveur DHCP ne pourra pas démarrer. Il suffit de créer un fichier vide. Pour cela taper la commande *touch /var/lib/dhcp3/dhcpd.leases*. Le fichier est créé. Voici ce que l’on peut avoir dedans après l’inscription du premier client:

```
[root@uranus /etc]# more  /var/lib/dhcp3/dhcpd.leases
```

```
lease 192.168.0.10 {
 starts 1 2002/12/14 18:33:45;
 ends 1 2002/12/14 18:34:22;
 hardware ethernet 00:40:33:2d:b5:dd;
 uid 01:00:40:33:2d:b5:dd;
 client-hostname "CHA100";
}
```

On distingue les informations suivantes : Début du bail, Fin du bail, adresse MAC du client, le nom d'hôte du client. Attention ce nom est différent du nom Netbios utilisé sur les réseaux Microsoft.

3.2.3. Activation du serveur

Le serveur est configuré, il n'y a plus qu'à le mettre en route. Utilisez les commandes suivantes:

- pour arrêter ou activer le service: */etc/init.d/dhcpd3 start | stop*

Le script lance le serveur en mode daemon. Vous pouvez le lancer en avant plan avec la commande "dhcpd3 -d". Cela permet de voir les messages et déterminer s'il y a des dysfonctionnement éventuels.

```
root@master:/etc/dhcp3# dhcpd3 -d
Internet Software Consortium DHCP Server V3.0.1rc9
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 1 leases to leases file.
Listening on LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   Socket/fallback/fallback-net
```

CTRL C pour arrêter.

3.3. Installation des clients

3.3.1. Le client sous Windows 9.x

L'installation est assez simple si vous avez déjà une carte réseau et le protocole TCP/IP installé. Utilisez les commandes suivantes: Panneau de configuration/Icône réseau/Protocole TCP IP/Propriétés/Onglet "adresse ip"/ Cochez "Obtenir automatiquement une adresse IP"

La configuration est terminée, vous pouvez relancer la machine. Le client interrogera un serveur DHCP pour qu'il lui délivre son autorisation de séjour sur le réseau.

3.3.2. Le client sous Linux

Vous allez réaliser une configuration manuelle

Allez dans le répertoire /etc/network, ouvrez le fichiers interfaces. C’est ici qu’est la configuration des cartes installées sur la machine. Remplacez "static" par "dhcp" dans la configuration de l’interface "eth0". Mettez tous les paramètres de cette interface (address, netmask, network....) en commentaire.

La configuration de la carte est terminée, vous pouvez tester en relançant le service réseau.

3.4. Procédure de test

Sur Windows vous allez pouvoir utiliser (enfin ça dépend des versions) les commandes IPCONFIG et Winipcfg.

Utilisez *ipconfig /?* pour voir comment utiliser la commande

Vous pouvez utiliser également "winipcfg". Allez dans Démarrer puis Exécuter et tapez *winipcfg*. Une fois la fenêtre activée vous pouvez utiliser les fonctions de libération et de renouvellement de bail. Si vous avez plusieurs cartes sur la station, la liste déroulante "Cartes Ethernet Informations" vous permet d’en sélectionner une.

4. TP

1. Installez un serveur DHCP minimal sous Linux et vérifiez le bon démarrage du service
2. Installez un client DHCP sous Linux, vérifiez le bon démarrage du service réseau et l’inscription dans le fichier dhcp.leases du serveur. Testez le renouvellement du baïl. Il suffit de relancer le service réseau.
3. Installez un client DHCP sous Windows, vérifiez le bon démarrage du service réseau et l’inscription dans le fichier dhcp.leases du serveur. Testez le renouvellement du baïl.
4. Modifiez l’étendue du serveur. Vérifiez le bon fonctionnement de la distribution d’adresses aux clients.
5. Modifiez la configuration du serveur afin qu’il distribue également l’adresse de la passerelle par défaut, l’adresse du serveur de nom. Testez la configuration.
6. Modifiez la configuration du serveur DHCP afin de réserver une adresse au client, vérifiez que le processus a bien fonctionné.

Installation d’un serveur DNS

Installation et configuration d’un serveur de nom.

1. Fiche de cours

1.1. Résumé

Ce document décrit la procédure d’installation et de configuration d’un serveur de noms sous GNU/Linux. Mots clés "Résolution de noms", "DNS", "NSLookup", "dig"

1.2. Description et objectifs de la séquence

Avant d’installer un service quel qu’il soit, il faut s’assurer du bon fonctionnement de la résolution de noms sur le réseau. Pour cela vous avez le choix entre l’utilisation des fichiers hosts ou du service DNS. C’est ce dernier qui sera utilisé. Vous devez être familiarisé avec l’installation de GNU/Linux.

1.3. Qu’est ce que le service de résolution de noms de domaine

Le service de résolution de noms d’hôtes DNS (Domain Name Services), permet d’adresser un hôte par un nom, plutôt que de l’adresser par une adresse IP. Quelle est la structure d’un nom d’hôte ?

```
              Nom_d_hôte  ou bien         Nom_d_hôte.NomDomaine
Exemple :     ns1 ou bien ns1.foo.org
```

Le nom de domaine identifie une organisation dans l’internet, comme, par exemple, yahoo.com, wanadoo.fr, eu.org. Dans les exemples, nous utiliserons un domaine que l’on considère fictif : "foo.org". Chaque organisation dispose d’un ou plusieurs réseaux. Ces réseaux sont composés de noeuds, ces noeuds (postes, serveurs, routeurs, imprimantes) pouvant être adressés.

Par exemple, la commande "ping ns1.foo.org", permet d’adresser la machine qui porte le nom d’hôte "ns1", dans le domaine (organisation) "foo.org".

Quelle différence entre la résolution de noms d’hôtes avec un serveur DNS et les fichiers "hosts" ?

Avec les fichiers "hosts", chaque machine dispose de sa propre base de données de noms. Sur des réseaux importants, cette base de données dupliquée n'est pas simple à maintenir.

Avec un service de résolution de noms, la base de données est localisée sur un serveur. Un client qui désire adresser un hôte regarde dans son cache local, s'il en connaît l'adresse. S'il ne la connaît pas il va interroger le serveur de noms.

Tous les grands réseaux sous TCP/IP, et internet fonctionnent (schématiquement) sur ce principe.

Avec un serveur DNS, un administrateur n'a plus qu'une seule base de données à maintenir. Il suffit qu'il indique sur chaque hôte, quelle est l'adresse de ce serveur. Ici il y a 2 cas de figures possibles :

- soit les hôtes (clients) sont des clients DHCP (Dynamic Host Configuration Protocol), cette solution est particulière et n'est pas abordée ici. Cette technique est l'objet d'un autre chapitre.
- soit les clients disposent d'une adresse IP statique. La configuration des clients est détaillée dans ce document.

Normalement un service DNS nécessite au minimum deux serveurs afin d'assurer un minimum de redondance. Les bases de données des services sont synchronisées. La configuration d'un serveur de noms secondaire sera expliquée. Nous verrons également en TP le fonctionnement de la réplication des bases de données (bases d'enregistrements de ressources). On peut parler de bases de données réparties et synchronisées.

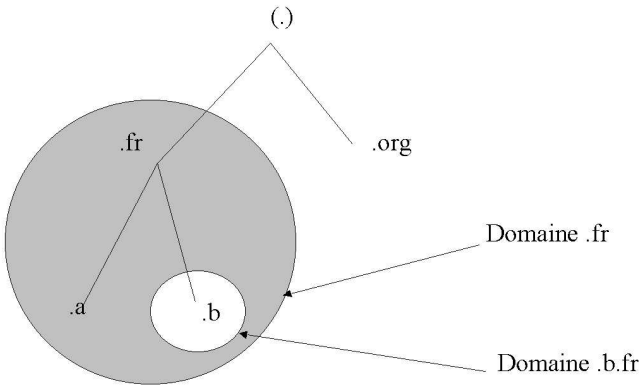
1.4. Présentation des concepts

1.4.1. Notion de domaine, de zone et de délégation

Un "domaine" est un sous-arbre de l'espace de nommage. Par exemple ".com" est un domaine, il contient toute la partie hiérarchique inférieure de l'arbre sous jacente au noeud ".com".

Un domaine peut être organisé en sous domaines. ".pirlouit.com" est un sous domaine du domaine ".com". Un domaine peut être assimilé à une partie ou sous-partie de l'organisation de l'espace de nommage. Voir la diapositive sur les Domaines, zones et délégations.

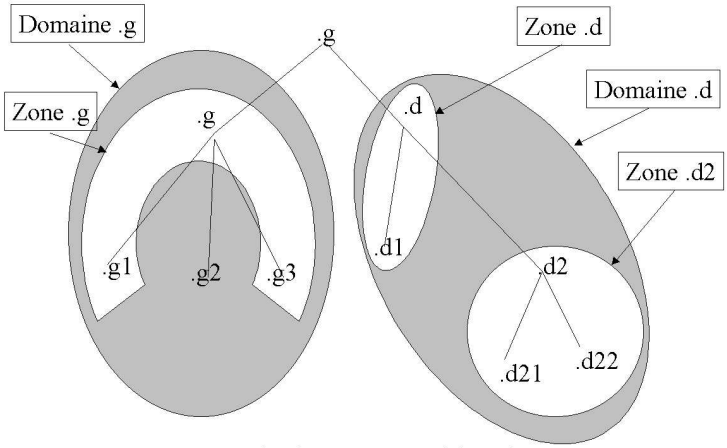
Figure 12. Les domaines



Un domaine est un sous arbre de l'espace de nommage.

Une "zone" est une organisation logique (ou pour être plus précis, une organisation administrative) des domaines. Le rôle d'une zone est principalement de simplifier l'administration des domaines. Le domaine ".com" peut être découpé en plusieurs zones, z1.com, z2.com...zn.com. L'administration des zones sera déléguée afin de simplifier la gestion globale du domaine. Voir la diapositive sur les zones.

Figure 13. Les zones

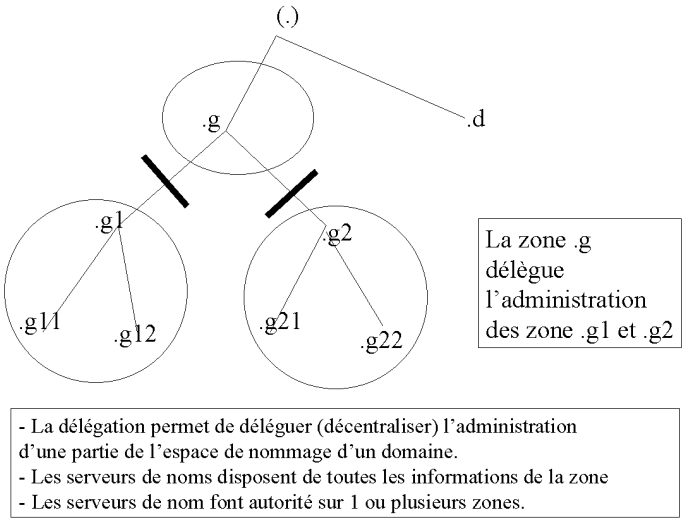


Une zone est une organisation gérée par délégation. C 'est un découpage en unités du domaine.

La délégation consiste à déléguer l'administration d'une zone (ou une sous-zone) aux administrateurs de cette

zone. Voir la diapositive sur la délégation.

Figure 14. La délégation



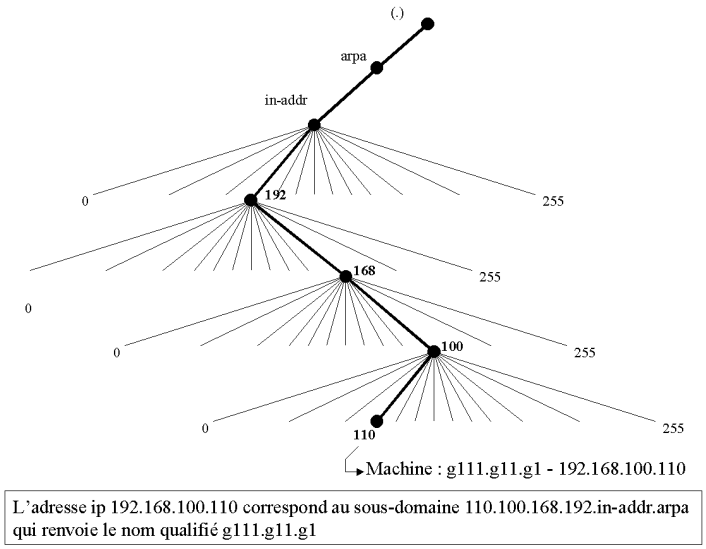
Attention à ces quelques remarques :

- Un domaine est une organisation de l'espace de nommage. Il peut être attaché à un domaine parent, et/ou peut avoir un ou plusieurs sous-domaines enfants.
- Les zones correspondent à des organisations administratives des domaines. Un domaine peut être administré par plusieurs zones administratives, mais il est possible aussi qu'une zone serve à l'administration de plusieurs domaines. Prenons l'exemple d'un domaine "MonEntreprise.fr", membre de ".fr". Il peut être composé de trois sous-domaines France.MonEntreprise.fr, Italie.MonEntreprise.fr, Espagne.MonEntreprise.fr et de deux zones d'administration. Une en France pour les sous-domaines France.MonEntreprise.fr, Italie.MonEntreprise.fr (il n'y a pas de délégation), et une pour Espagne.MonEntreprise.fr, il y a délégation.
- L'adressage IP correspond à une organisation physique des noeuds sur un réseau IP.
- L'organisation de l'espace de nommage est complètement indépendante de l'implantation géographique d'un réseau ou de son organisation physique. L'organisation physique est gérée par des routes (tables de routage). L'espace de nommage indique pour un nom de domaine N, quelles sont les serveurs de noms qui ont autorité sur cette zone. Elles ne donnent pas la façon d'arriver à ces machines.
- Les seules machines connues au niveau de l'espace de nommage, sont les serveurs de nom "déclarés". Ces informations sont accessibles par des bases de données "whois".
- La cohérence (le service de résolution de noms) entre l'organisation de l'espace de nommage global et les organisations internes des réseaux sur internet est réalisée par les serveurs de noms.

1.4.2. le domaine in-addr.arpa

Le principe de la résolution de noms, consiste à affecter un nom d'hôte une adresse IP. On parle de résolution de noms directe. Le processus inverse doit pouvoir également être mis en oeuvre. On parle de résolution de noms inverse ou reverse. Le processus doit fournir, pour une adresse IP, le nom correspondant. Pour cela il y a une zone particulière, in-addr.arpa, qui permet la résolution inverse d'adresse IP. Voir la diapositive sur la résolution inverse.

Figure 15. La résolution inverse



Par exemple, pour le réseau 192.68.1.0, on créera une zone inverse dans le domaine in-addr.arpa. La zone de recherche inverse dans le domaine deviendra : 1.68.192.in-addr.arpa. Cette zone devra répondre pour toutes les adresses déclarées dans la tranche 192.168.1.0 à 192.168.1.254.

On inscrira dans cette zone tous les noeuds du réseau pour lesquels on désire que la résolution inverse fonctionne. Un serveur de noms peut, pratiquement, fonctionner sans la définition de cette zone tant que le réseau n'est pas relié à l'internet. Si cela était le cas, il faudrait déclarer cette zone, sans quoi, des services comme la messagerie électronique, ne pourrait fonctionner correctement, notamment à causes des règles anti-spam. (Voir www.nic.fr)

1.4.3. Fichiers, structure et contenus

Sur linux nous allons utiliser deux types de fichiers :

- le fichier /etc/bind/named.conf, qui décrit la configuration générale du serveur DNS,
- les fichiers qui contiennent les enregistrements de ressources pour la zone dans /etc/bind. On crée, en général, un fichier pour la résolution directe d'une zone, et un fichier pour la résolution inverse.

Les enregistrements ont une structure et un rôle que nous verrons. Le daemon se nomme "named", prononcer "naime dé".

1.4.4. Principaux types d’enregistrements

Les types d’enregistrements qui enrichissent une base de données DNS sont de plusieurs types, dont voici les principaux:

- Enregistrement de type SOA (Start Of Authority) : Indique l’autorité sur la zone. Ces enregistrements contiennent toutes les informations sur le domaine. Par exemple le délai de mise à jour des bases de données entre serveurs de noms primaires et secondaires, le nom du responsable du site
- Enregistrements de type NS (Name Server) : Ces enregistrements donnent les adresses des serveurs de noms pour le domaine.
- Enregistrement de type A (Adresse) : Ces enregistrements permettent de définir les noeuds fixes du réseau (ceux qui ont des adresses IP statiques). Serveurs, routeurs, switchs ...
- Enregistrements de type MX (Mail eXchanger) : Ils servent pour déclarer les serveurs de messagerie. Il faudra déclarer une enregistrement de type "MX" pour la réalisation du TP sur la messagerie.
- Enregistrements de type CNAME (Canonical Name) : Ils permettent de définir des alias sur des noeuds existants. Par exemple www.foo.org peut être la même machine que web.foo.org. Dans ce cas, "www" est un alias (CNAME) de "web". Cela permet de différencier le nommage des machines des standards de nommages des services (www, ftp, news, smtp, mail, pop...).
- Enregistrement de type PTR (Pointeur) : Ils permettent la résolution de noms inverse dans le domaine in-addr.arpa.

Ces enregistrements caractérisent des informations de type IN - INternet. Voir l’annexe pour avoir un fichier exemple.

1.4.5. Structure des enregistrements

Structure d’un enregistrement SOA : Chaque fichier de ressource de zone, commence par un enregistrement de type SOA. Voici un exemple d’enregistrement SOA.

```
$TTL 38400
foo.org. IN SOA ns1.foo.org. hostmaster.foo.org. (
20001210011     ; numéro de série
10800           ; rafraîchissement
3600            ; nouvel essai
604800          ; Obsolescence après une semaine
86400 )         ; TTL minimal de 1 jour
```

Caractéristiques des différentes informations :

SOA Start Of Authority, enregistrement qui contient les informations de synchronisation des différents serveurs de nom.

foo.org, donne le nom de la zone. Le nom de la zone, ici "foo.org", peut être remplacé par "l'@", arobase.

hostmaster.foo.org : la personne qui est responsable de la zone. Le premier point sera remplacé par l'arobase (@) pour envoyer un courrier électronique. Cela deviendra hostmaster@foo.org. En général postmaster, est un alias de messagerie électronique vers l'administrateur du DNS.

1. Numéro de série sous la forme AAAAMMJJNN, sert à identifier la dernière modification sur le serveur de noms maître. Ce numéro sera utilisé par les serveurs de nom secondaires pour synchroniser leurs bases. Si le numéro de série du serveur de noms primaire est supérieur à celui des serveurs de noms secondaire, alors le processus de synchronisation suppose que l'administrateur à apporté une modification sur le serveur maître et les bases sont synchronisées.
2. Rafraîchissement : Intervalle de temps donné en seconde pour indiquer au serveur la périodicité de la synchronisation.
3. Retry : Intervalle de temps avant réitération si l'essai précédent n'a pas fonctionné.
4. Expire : Temps au bout duquel le serveur ne remplit plus sa mission s'il n'a pu contacter le serveur maître pour mettre à jour ses données.
5. TTL : Time To Live, durée de vie des enregistrements. Plus la durée de vie est courte, plus l'administrateur est susceptible de considérer que ses bases sont à jour, par contre cela augmente le trafic sur le réseau.

Enregistrement de type NS pour le domaine foo.org:

```
foo.org. IN NS ns1.foo.org. ; noter le point final "."
foo.org. IN NS ns2.foo.org. ; foo.org peut être remplacé par "@"
; IN signifie enregistrement de type INternet
```

Le "." final signifie que le nom est pleinement qualifié. On aurait pu mettre :

```
@ IN NS ns1
IN NS ns2
```

"@" signifie "foo.org" et pour le serveur de nom, comme "ns1" n'est pas pleinement qualifié, cela équivaut à "ns1.foo.org".

Enregistrements de type A : Nous devons décrire la correspondance Nom / Adresse

```
ns1.foo.org. IN A 192.168.0.1
ns2.foo.org. IN A 192.168.0.2
localhost.foo.org. IN A 127.0.0.1
```

S'il y avait d'autres hôtes sur la zone, il faudrait les définir ici.

Enregistrements de type CNAME : Ce sont les alias (Canonical Name). Une requête du type http://www.foo.org sera adressée à ns1.foo.org, puisque www est un alias de ns1.

```
www IN CNAME ns1.foo.org.
ftp IN CNAME ns1.foo.org.
```

Enregistrement de type PTR : Il serviront à la résolution de noms inverse.

```
1.0.168.192.in-addr.arpa. IN PTR ns1.foo.org.
2.0.168.192.in-addr.arpa. IN PTR ns2.foo.org.
```

1.4.6. La délégation

La délégation consiste à donner l’administration d’une partie du domaine à une autre organisation. Il y a transfert de responsabilité pour l’administration d’une zone. Les serveurs de la zone auront autorité sur la zone et auront en charge la responsabilité de la résolution de noms sur la zone. Les serveurs ayant autorité sur le domaine auront des pointeurs vers les serveurs de noms ayant autorité sur chaque zone du domaine.

1.4.7. Serveur primaire et serveur secondaire

Le serveur maître (primaire) dispose d’un fichier d’information sur la zone. Le ou les serveurs esclaves (secondaires) obtiennent les informations à partir d’un serveur primaire ou d’un autre serveur esclave. Il y a " transfert de zone". Les serveurs maîtres et esclaves ont autorité sur la zone.

1.4.8. Le cache

L’organisation d’internet est assez hiérarchique. Chaque domaine dispose de ses propres serveurs de noms. Les serveurs peuvent être sur le réseau physique dont ils assurent la résolution de nom ou sur un autre réseau. Chaque zone de niveau supérieur (edu, org, fr...) dispose également de serveurs de nom de niveau supérieur. L’installation du service DNS, installe une liste de serveurs de noms de niveaux supérieurs. Cette liste permet au serveur de résoudre les noms qui sont extérieurs à sa zone. Le serveur enrichit son cache avec tous les noms résolus. Si votre réseau réseau n’est pas relié à internet, vous n’avez pas besoin d’activer cette liste.

Ce fichier est un peu particulier. Il est fournit avec les distributions. Il est utilisé par le serveur de noms à l’initialisation de sa mémoire cache. Si vos serveurs sont raccordés à internet, vous pourrez utiliser une liste officielle des serveurs de la racine (ftp.rs.internic.net).

1.5. Installation et confi guration d’un serveur DNS

Processus de configuration

L’application est déjà installée. Pour mettre en place le service de résolution de noms sur un serveur GNU/Linux, on va procéder successivement aux opérations suivantes :

1. vérifier les fichiers déjà installés,
2. configurer les fichiers des zones administrées,
3. configurer les fichiers de transaction sécurisée pour rndc,

4. démarrer et tester le service serveur.

1.5.1. Fichiers déjà installés

Vous devez normalement avoir déjà les fichiers suivants :

1. /etc/bind/named.conf, fichier de déclaration des fichiers de ressources
2. /etc/bind/db.127, zone locale reverse
3. /etc/bind/db.0, zone locale de broadcast
4. /etc/bind/db.255, zone locale de broadcast
5. db.local, zone directe locale
6. db.root, fichiers des serveurs racine

Le contenu de tous ces fichiers et commentaires se trouve en annexe.

Vous avez également des fichiers particuliers : "rndc.key", "rndc.conf". rndc, est un outil qui permet de passer des commandes à distance à un serveur de nom. Nous porterons une attention toute particulière à ces fichiers, à leur rôles et à l'utilité de rndc.

Il va suffire de rajouter les fichiers manquants à la zone administrée.

1.5.2. rndc, le fichier de configuration, le fichier de clé

rndc est un outil qui permet de réaliser des transactions sécurisées avec un serveur de nom. Le mode de fonctionnement est dit à "clé partagée", c'est à dire que le client rndc et le serveur bind doivent avoir la même clé. Vous devrez donc configurer le fichier de configuration de rndc et le fichier named.conf avec les mêmes paramètres.

Ces fichiers et exemples sont également fournis en annexe. La clé doit être strictement identique dans les 2 fichiers. Si vous avez un message d'erreur à l'utilisation de rndc, vérifiez bien ces paramètres.

rndc supporte plusieurs paramètres pour passer des commandes au serveur de nom (halt, querylog, refresh, reload, stat...). Utilisez la commande "man rndc" pour en savoir plus.

Dans le fichier rndc, vous allez avoir besoin d'au moins 3 paramètres. rndc utilisera ces paramètres si rien n'est spécifié sur la ligne de commande. Dans les autres cas, vous pouvez passer les paramètres sur la ligne de commande.

Note : vous pouvez vous passer du système de clé mais ce n'est pas conseillé. Commentez tout ce qu'il y a dans le fichier named.conf et qui concerne la clé s'il y a déjà des choses. Renommez le fichier rndc.conf en

rndc.conf.orig, ça devrait fonctionner. Vous pouvez tester cela en faisant un "/etc/init.d/bind restart". Vous ne devriez pas avoir de message d’erreur.

```
#Description du serveur et de la clé utilisés par défaut.
#Ici on utilise par défaut le serveur local, avec la clé key-name
   options {
     default-server  localhost;
     default-key     "<key-name>";
   };
```

Il est possible de dire quelle clé utiliser en fonction d’un serveur donné.

```
   server localhost {
     key  "<key-name>";
   };
```

Enfin il reste à définir la ou les clés avec leur noms et leurs valeurs.

```
   key "<key-name>" {
     algorithm hmac-md5;
     secret "<key-value>";
   };
```

Pour créer une nouvelle clé, utilisez la commande :

```
dnssec-keygen -a hmac-md5 -b <bit-length> -n HOST <key-file-name>

#Ici on génère une clé de 512 bits dans un fichier maCLE
dnssec-keygen -a hmac-md5 -b 512 -n HOST maCLE
```

Le fichier named.conf doit connaître la clé utilisée par le client,

```
// secret must be the same as in /etc/rndc.conf
key "key" {
        algorithm       hmac-md5;
        secret
"c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
};
```

mais doit également comprendre les paramètres qui définissent les machines clientes autorisées à passer des commandes avec une directive "controls".

```
controls {
     inet 127.0.0.1 allow { localhost; } keys { <key-name>; };
};

# Ici on peut passer des commandes à partir de n’importe quelle machine
controls {
    inet 127.0.0.1 allow { any; } keys { "key"; };
};

# Ici on peut passer des commandes localement
```

```
controls {
    inet 127.0.0.1 allow { localhost; } keys { "key"; };
};
```

1.5.3. Procédure de configuration du serveur

L'installation a copié les fichiers. Sur une configuration simple vous allez avoir 3 fichiers à créer ou à modifier sur le serveur primaire :

- /etc/bind/named.conf (fichier de configuration globale du service DNS du serveur de noms primaire),
- /etc/bind/db.foo.org qui contiendra la description de la correspondance nom-adresse de toutes les machines du réseau
- /etc/bind/db.foo.org.rev qui contiendra la correspondance inverse adresse-nom (pour la résolution inverse de nom in-addr.arpa).

1.5.4. Configurer les fichiers

Vous pouvez configurer le serveur manuellement, c'est à dire créer les fichiers à l'aide d'un éditeur de texte ou à l'aide d'un outil de configuration graphique. En général on n'installe jamais d'interface graphique sur un serveur pour des questions de sécurité. Nous allons donc créer les fichiers complètement. La configuration est réalisable également à distance avec des requêtes HTTP grâce à des outils comme "webmin".

1.5.5. Configuration du DNS manuellement

Le fichier racine pour la configuration du serveur de noms est le fichier "/etc/bind/named.conf". Ce fichier est lu au démarrage du service et donne la liste des fichiers qui définissent la base de données pour la zone.

1.5.6. Le fichier named.conf

Voir annexe.

1.5.7. Le fichier db.foo.org

Voir annexe.

Le paramètre @, signifie qu'il s'agit du domaine "foo.org" (le nom tapé après le mot " zone " dans le fichier de configuration named.conf). Le paramètre "IN", signifie qu'il s'agit d'un enregistrement de type internet. Notez la présence d'un point (.) après le nom des machines pleinement qualifiés. Sans celui-ci, le nom serait " étendu ".

Par exemple, ns1.foo.org (sans point) serait compris comme ns1.foo.org.foo.org (on rajoute le nom de domaine en l'absence du point terminal). Le point (.) terminal permet de signifier que le nom est pleinement qualifié.

1.5.8. Le fichier db.foo.org.rev

Voir annexe.

1.6. Compléments pratiques

1.6.1. Démarrer ou arrêter le service le service

Le service (daemon) qui active la résolution de noms s'appelle "named", prononcer "naime dé", mais le script s'appelle "bind", ou sur certaines distributions "bind9". Je noterai ici "bind".

Si vous voulez l'arrêter ou le redémarrer dynamiquement vous pouvez utiliser les commandes suivantes :

```
# La commande stop utilise souvent rndc.
# rndc doit donc être préalablement configuré.
/etc/init.d/bind stop
/etc/init.d/bind start
```

Relancer le service serveur de cette façon peut parfois poser problème. En effet cette procédure régénère le cache du serveur. Le service prend également un nouveau "PID". Si vous voulez éviter cela, ce qui est généralement le cas, préférez la commande "kill -HUP 'PID de Named'". Vous trouverez le PID de named dans "/var/run".

1.6.2. Finaliser la configuration

Les fichiers de configuration sont créés. Il ne reste plus qu'à tester. Il faut au préalable configurer le serveur pour que tous les processus clients utilisent le service de résolution de nom. Il vous faut modifier le fichier "/etc/resolv.conf":

```
# nameserver AdresseIpDuServeurDeNom
# Exemple
nameserver 192.168.0.1
```

Vous pouvez également configurer d'autres clients pour qu'ils utilisent votre serveur de nom.

1.6.3. Procédure de configuration des clients

La description de la configuration de tous les clients possibles n'est pas détaillée. Vous trouverez ci-dessous des éléments pour un client windows 9x et pour un client GNU/Linux.

1.6.4. Avec windows

Il s’agit d’un client windows. Chaque client dispose du protocole TPC/IP, d’une adresse IP. Il faut configurer le client pour lui signifier quel est le serveur de noms qu’il doit consulter. Pour cela il faut aller dans : panneau de configuration - réseau - tcp/ip - Onglet "Configuration DNS". Vous allez pouvoir définir les paramètres suivants

- le nom d’hôte de la machine locale dans le réseau
- le nom de domaine auquel appartient l’hôte (dans cet exemple c’est foo.org)

Ces 2 paramètres sont facultatifs dans l’atelier qui nous intéresse. Par contre le paramètre "Ordre de recherche DNS" est important. Mettez dessous :

- L’adresse IP du serveur de noms que vous avez configuré,
- Cliquez sur ajouter
- Entrez l’adresse IP du serveur de noms
- Validez puis relancer la machine

Ce paramètre, définit à la machine locale, l’adresse de l’hôte de destination qui est chargé de la résolution des noms d’hôtes dans le réseau. Cela permet de dire qu’un serveur de noms doit avoir une adresse IP statique sur le réseau.

1.6.5. Avec GNU/Linux

Vous pouvez modifier (en tant que " root ") le fichier de configuration du " resolver " (/etc/resolv.conf). Exemple (ça tient en deux lignes) :

```
# Fichier /etc/resolv.conf
search foo.org
nameserver 192.168.1.1 # mettre votre DNS
```

1.7. Procédure de tests

Attention aux fichier "hosts" et au fichier "host.conf". Prenez le temps de regarder ce qu’il y a dedans. Faites une copie de sauvegarde de ces fichiers et renommez les. Vérifiez au besoin leur utilité avec les commandes "man host.conf" et "man hosts".

Vous pouvez tester votre configuration avant même d’avoir configuré un client. Sur la même machine vous allez utiliser un service client du serveur (commande ping) qui utilisera un service serveur (DNS).

Test sur le serveur de noms : Tapez la commande " ping ftp.foo.org ". Si la commande répond, le serveur fonctionne. En effet ftp est un alias de ns1 dans la zone foo.org.

Test sur le client : Avant de lancer une commande, vous devez vérifier que vous n’avez pas de fichier "hosts" local, sinon vous devez le supprimer.

Pourquoi ? L’utilisation de fichiers hosts et d’un serveur de noms n’est pas exclusif. Dans bien des environnements, le fichier hosts est consulté avant le serveur de noms (notamment windows, GNU/Linux à moins que ce ne soit précisé). Si vous avez un fichier hosts sur la machine, vous pouvez avoir des résultats qui ne sont pas ceux attendus.

1.7.1. Vérifi er la résolution de noms :

Pensez à bien vérifier le nom d’hôte de votre machine avec la commande "hostname", au besoin, sous root, modifiez ce nom, toujours avec cette commande. Fermez les sessions et rouvrez les, vous aurez le bon nom d’hôte qui s’affichera sur votre console.

Mettons que le réseau soit configuré de la façon suivante:

```
Nom d’hôte Alias (CNAME) Adresse IP Serveur
ns1 www
ftp
mail
ns1 192.68.1.1
Client 1 Cli1 192.68.1.2
```

Pour vérifier le fonctionnement de la résolution de noms à partir du client cli1, vous pouvez utiliser les commandes suivantes :

```
- ping ns1
- ping cli1
```

Vous pouvez également tester la résolution des alias (CNAME) avec les commandes :

```
ping mail.foo.org
ping www.foo.org
ping ftp.foo.org
ping ns1.foo.org
```

C’est bien la même adresse IP (voir le cache arp) qui répond, la machine a donc bien plusieurs noms.

Si vous voulez vérifier que c’est bien le serveur de noms qui réalise la résolution, il existe plusieurs solutions. La plus simple est d’arrêter le service serveur avec la commande : /etc/init.d/bind stop, puis de refaire les manipulations. Aucune machine n’est atteignable en utilisant son nom, mais cela est toujours possible en utilisant l’adresse IP.

1.8. Dépannage et outils

Les sources de dysfonctionnement des services de nom peuvent être nombreuses et parfois complexes à résoudre. Voici quelques outils et méthodes qui peuvent être utilisées.

1.8.1. Les erreurs de chargement de bind

Si vous avez une erreur similaire à celle-ci :

```
Problème de clés entre named et rndc
root@knoppix:/etc/bind# /etc/init.d/bind9 stop
Stopping domain name service: namedrndc: connection to remote host closed
This may indicate that the remote server is using an older version of
the command protocol, this host is not authorized to connect,
or the key is invalid.
```

Le problème est lié à rndc, et souvent à des clés qui sont différentes ou mal définies entre named.conf et rndc.conf. Vérifiez donc bien tous les paramètres.

Vérifiez dans les journaux (en général /var/log/daemon) qu'il n'y a pas d'erreur de chargement de named. Voici un exemple de log.

```
# Log après chargement des zones
Apr  8 23:12:46 knoppix named[1066]: starting BIND 9.2.1
Apr  8 23:12:46 knoppix named[1066]: using 1 CPU
Apr  8 23:12:46 knoppix  []

named[1068]: loading configuration from '/etc/bind/named.conf'
named[1068]: no IPv6 interfaces found
named[1068]: listening on IPv4 interface lo, 127.0.0.1#53
named[1068]: listening on IPv4 interface eth0, 192.168.90.100#53
named[1068]: command channel listening on 127.0.0.1#953
named[1068]: zone 0.in-addr.arpa/IN: loaded serial 1
named[1068]: zone 127.in-addr.arpa/IN: loaded serial 1
named[1068]: zone 255.in-addr.arpa/IN: loaded serial 1
named[1068]: zone localhost/IN: loaded serial 1
named[1068]: zone foo.org/IN: loaded serial 2003040101
Apr  8 23:12:46 knoppix named[1068]: running
```

Ou encore avec la commande ps :

```
root:# ps aux | grep named
root  1066  0.0  1.6 10312 2136 ?  S    23:12   0:00 /usr/sbin/named
root  1067  0.0  1.6 10312 2136 ?  S    23:12   0:00 /usr/sbin/named
root  1068  0.0  1.6 10312 2136 ?  S    23:12   0:00 /usr/sbin/named
root  1069  0.0  1.6 10312 2136 ?  S    23:12   0:00 /usr/sbin/named
root  1070  0.0  1.6 10312 2136 ?  S    23:12   0:00 /usr/sbin/named
```

Vous pouvez également faire des tests successifs pour tester la résolution de nom.

```
#Vérification avec des ping
```

```
root@ns1:~# ping -c1 ns1.foo.org
PING ns1.foo.org (192.168.90.100): 56 data bytes
64 bytes from 192.168.90.100: icmp_seq=0 ttl=64 time=0.1 ms
--- ns1.foo.org ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms


root@ns1:~# ping -c1 www.foo.org
PING ns1.foo.org (192.168.90.100): 56 data bytes
64 bytes from 192.168.90.100: icmp_seq=0 ttl=64 time=0.1 ms
--- ns1.foo.org ping statistics ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.1 ms
```

1.8.2. nslookup, dig

La commande " nslookup " est de moins en moins utilisée, nous la verrons donc pas. Nous allons voir l'utilisation de dig.

Ces commandes sont très largement utilisées par les administrateurs de réseau pour résoudre les problèmes liés aux services de résolution de nom.

Tests avec dig :

```
# Test sur une zone
root@knoppix:/etc/bind# dig any foo.org
; <<>> DiG 9.2.1 <<>> any foo.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32752
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; QUESTION SECTION:
;foo.org.                       IN      ANY

;; ANSWER SECTION:

foo.org.                604800  IN      SOA     ns1.foo.org.   \
        root.ns1.foo.org. 2003040102 604800 86400 2419200 604800
foo.org.                604800  IN      NS      ns1.foo.org.

;; ADDITIONAL SECTION:
ns1.foo.org.            604800  IN      A       192.168.90.100

;; Query time: 7 msec
;; SERVER: 192.168.90.100#53(192.168.90.100)
;; WHEN: Tue Apr  8 23:30:05 2003
;; MSG SIZE  rcvd: 100
```

```
# Récupération de l'enregistrement SOA d'une zone

root@knoppix:/etc/bind# dig soa foo.org

; <<>> DiG 9.2.1 <<>> soa foo.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 15982
;; flags: qr aa rd ra; QUERY:1, ANSWER:1, AUTHORITY:1, ADDITIONAL:1

;; QUESTION SECTION:
;foo.org.                       IN      SOA

;; ANSWER SECTION:

foo.org.                604800  IN      SOA     ns1.foo.org.     \
          root.ns1.foo.org. 2003040102 604800 86400 2419200 604800

;; AUTHORITY SECTION:
foo.org.                604800  IN      NS      ns1.foo.org.

;; ADDITIONAL SECTION:
ns1.foo.org.            604800  IN      A       192.168.90.100

;; Query time: 2 msec
;; SERVER: 192.168.90.100#53(192.168.90.100)
;; WHEN: Tue Apr  8 23:30:43 2003
;; MSG SIZE  rcvd: 100


#Vérification de la résolution de nom sur www.foo.org

root@knoppix:/etc/bind# dig www.foo.org

; <<>> DiG 9.2.1 <<>> www.foo.org
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 52961
;; flags: qr aa rd ra; QUERY:1, ANSWER:2, AUTHORITY:1, ADDITIONAL:0

;; QUESTION SECTION:
;www.foo.org.                   IN      A

;; ANSWER SECTION:
www.foo.org.            604800  IN      CNAME   ns1.foo.org.
ns1.foo.org.            604800  IN      A       192.168.90.100

;; AUTHORITY SECTION:
foo.org.                604800  IN      NS      ns1.foo.org.

;; Query time: 3 msec
;; SERVER: 192.168.90.100#53(192.168.90.100)
;; WHEN: Tue Apr  8 23:31:49 2003
;; MSG SIZE  rcvd: 77
```

```
# Vérification de la résolution de nom inverse.

root@ns1:/etc/bind# dig ptr 100.90.168.192.in-addr.arpa
; <<>> DiG 9.2.1 <<>> ptr 100.90.168.192.in-addr.arpa
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30642
;; flags: qr aa rd ra; QUERY:1, ANSWER:1, AUTHORITY:1, ADDITIONAL:0
;; QUESTION SECTION:
;100.90.168.192.in-addr.arpa.   IN      PTR

;; ANSWER SECTION:
100.90.168.192.in-addr.arpa. \
                  604800 IN  PTR  ns1.90.168.192.in-addr.arpa.

;; AUTHORITY SECTION:
90.168.192.in-addr.arpa.     \
                  604800 IN      NS      ns1.90.168.192.in-addr.arpa.

;; Query time: 7 msec
;; SERVER: 192.168.90.100#53(192.168.90.100)
;; WHEN: Tue Apr  8 23:45:39 2003
;; MSG SIZE  rcvd: 77
```

1.8.3. Le cache du DNS

Le cache permet également de détecter certaines causes d’erreur. Le problème est qu’il est en mémoire. Pour le récupérer sous la forme d’un fichier utilisez la commande : "kill -INT ’PID de named’". Vous récupérez un fichier /var/named/named_dump.db que vous pouvez exploiter.

1.8.4. Les journaux

Si vous êtes en phase de configuration, pensez (ce doit être un réflexe) à consulter les fichiers de journalisation, notamment "/var/log/messages". Cette opération permet dans bien des cas de corriger des erreurs qui se trouvent dans les fichiers de configuration. Voici comment procéder:

- Arrêt du serveur
- Nettoyage du fichier "> /var/log/messages"
- Démarrage du serveur
- Consultation des logs : cat /var/log/daemon.log | more

Pour les fichiers logs, utilisez, si le fichier est trop gros la commande "tail" :

```
# tail -N NomFichier
# Affiche les N dernières lignes d’un fichier
# Par exemple, affiche les 250 dernières lignes d’un fichiers
# tail -n 250 /var/log/daemon.log | more
```

1.9. Remarques

Si vous désirez mettre en place la résolution de noms sur un réseau local, il n'y a pas grand chose de plus à réaliser. Il faut rajouter les enregistrements de type MX pour la messagerie, cette opération sera réalisée pendant la configuration du service de messagerie. Il faut également mettre en place un service de synchronisation des bases de données avec un serveur secondaire pour assurer le service d'un serveur de noms de backup.

Si vous désirez vous relier sur internet, le processus est plus complexe. Il faudra approfondir la description des enregistrements et la structure des fichiers.

Par convention, on considère que chaque domaine dispose d'au moins 1 serveur de noms primaire et un serveur de noms secondaire afin d'assurer une redondance en cas de panne d'un serveur. Les clients réseau seront configurés pour utiliser indifféremment le serveur de noms primaire ou les serveurs de nom secondaires. Il en résulte une duplication de la base de données du DNS primaire sur les serveurs secondaires. La base de données est rafraîchie en fonction des paramètres de l'enregistrement SOA. Ce procédé met en oeuvre un principe de base de données répartie. Vous trouverez quelques éléments dans les annexes qui suivent.

1.10. Annexes

1.10.1. Annexe 1 - Extraits de fichiers de configuration

Les extraits ci-dessous d'une zone fictive foo.org peuvent servir d'exemple pour bâtir une zone.

Si on respecte les conventions utilisées sur internet, voici ce que l'on devrait avoir :

- le serveur ftp est accessible par l'adresse ftp.foo.org
- le serveur http par l'adresse www.foo.org
- le serveur de noms primaire par ns1.foo.org
- le serveur de messagerie mail.foo.org
- le serveur de news news.foo.org, etc, etc.

ftp, www, mail sont des alias (canonical name ou CNAME) de la machine " ns1 " dans le domaine foo.org

Nous aurons donc sur le serveur de noms 5 enregistrements dans la zone foo.org qui concernent la machine ns1.foo.org : un enregistrement de type A pour déclarer ns1 quatre enregistrements de type CNAME pour la machine ns1.

Nous aurons également, dans la zone reverse in-addr.arpa, 1 enregistrement de type pointeurs (PTR) pour chaque enregistrement de type A dans la zone directe. Enfin, pour le serveur de messagerie, il faut également un enregistrement de type MX.

Tous les fichiers concernant la zone locale, et un fichier named.conf sont déjà installés sur votre machine.

```
; db.local
; Résolution directe pour la zone locale
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
@       IN      A       127.0.0.1



; db.127
; Résolution inverse pour l'adresse de loopback
; BIND reverse data file for local loopback interface
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.
1.0.0   IN      PTR     localhost.



; db.0
; Résolution inverse pour la zone de broadcast
; BIND reverse data file for broadcast zone
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.



; db.255
; Résolution inverse pour la zone de broadcast;
; BIND reverse data file for broadcast zone
;
$TTL    604800
@       IN      SOA     localhost. root.localhost. (
                              1         ; Serial
```

```
                        604800         ; Refresh
                         86400         ; Retry
                       2419200         ; Expire
                        604800 )       ; Negative Cache TTL
;
@       IN      NS      localhost.


; db.root
; fichier des serveurs de noms de l'internet
; vous pouvez le consulter sur votre disque.


; db.foo.org
; fichier directe pour la zone foo.org
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1 root.ns1 (
                     2003040102         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1
ns1     IN      A       192.168.90.100 ; @ ip du serveur de nom
www     IN      CNAME   ns1
ftp     IN      CNAME   ns1


; db.foo.org.rev
; fichier de résolution inverse pour la zone foo
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     ns1 root.ns1 (
                     2003040102         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      ns1
100     IN      PTR     ns1



; fichier named.conf du serveur primaire
// C'est le fichier principal de configuration des DNS
// C'est ici que sont réalisées„ pour chaque zone, les déclarations
// des fichiers de ressources.

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

```
// Serveurs à prévenir pour les transferts de zone
        forwarders {0.0.0.0;};
};

// Ici les paramètres pour les clés rndc
// Les paramètres doivent être strictement identiques à celui
// du fichier rndc.key ou rndc.conf
// Si vous avez des messages d'erreur à l'utilisation
// de cette commande, vérifier le contenu des fichiers.


key "rndc-key" {
algorithm       hmac-md5;
secret "c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
};

# Autoriations rndc sur la machine.
controls {
   inet  127.0.0.1 allow {localhost;} keys {"rndc-key";};
};

// Indication pour les serveurs racines
zone "." {
        type hint;
        file "/etc/bind/db.root";
};


// be authoritative for the localhost forward
// and reverse zones, and for
// broadcast zones as per RFC 1912
zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

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

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

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


//zone directe de foo
zone "foo.org" {
        type master;
        file "/etc/bind/db.foo.org";
};
```

```
//Zone reverse pour 192.168.90.
zone "90.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/db.foo.org.rev";
};



; fichier name.conf du serveur secondaire
; L'entête de bouge pas
; Tout ce qui concerne localhost non plus car chaque DNS
; est primaire pour les zones locales
; on ajoute la déclarations des autres zones, le fichier
; de stockage et l'adresse IP du serveur primaire pour
; pouvoir réaliser les transferts de zone.

; Déclaration de la zone foo.org

zone "foo.org" {
        type slave;
        file "/etc/bind/db.foo.org";
        masters {192.168.90.1;};
};

zone "90.168.192.in-addr.arpa" {
        type slave;
        file "/etc/bind/db.foo.org.rev";
        masters {192.168.90.1;};
};



; fichier rndc.conf
/* $Id: rndc.conf,v 1.1 2001/02/13 07:15:34 bdale Exp $ */
/*
 * Exemple de fichier de rndc.conf, pris pour les TP
 */

options {
        default-server  localhost;
        default-key     "rndc-key";
};

server localhost {
        key     "rndc-key";
};

key "rndc-key" {
algorithm       hmac-md5;
secret "c3Ryb25nIGVub3VnaCBmb3IgYSBtYW4gYnV0IG1hZGUgZm9yIGEgd29tYW4K";
};



; fichier rndc.key

key "rndc-key" {
```

```
        algorithm hmac-md5;
        secret "49khYQyHfO4AqYO9K7by6Q==";
};
```

1.10.2. Annexe 2 - Serveur primaire et serveur secondaire

Pour configurer le serveur secondaire, vous n'avez pas grand chose à faire. Copiez le fichier named.conf du primaire sur le secondaire. Voyez l'exemple ci-dessus. Le dns secondaire téléchargera (processus de transfert de zone) les fichiers de ressources du dns primaire. Attention, le dns secondaire pour une zone est toujours dns primaire pour la zone locale "localhost".

On remplace la définition "masters" par "slave" sauf pour la zone locale et les fichiers db.local et db.127 qui sont lus localement. Ensuite vous avez rajouté l'adresse du serveur à partir duquel le transfert de zone doit s'effectuer.

Activer les serveurs de noms et analyser les traces (log) sur les 2 serveurs. Corrigez toutes les erreurs jusqu'à ce que cela fonctionne. Vous devriez obtenir la trace selon laquelle il y a eu un transfert de zone entre le serveur maître et le serveur esclave. Exemple :

```
Apr  6 plibre named[8821]: send AXFR query 0 to 195.115.88.38
```

*Expérience 1 :*Vous pouvez expérimenter un échange entre un serveur de noms primaire et un serveur esclave. Modifiez sur le serveur primaire le N° de série comme si vous aviez modifié les fichiers de ressources de ns1 et relancez le service. Vérifiez le transfert de zone a mis à jour la base de données répartie.

Expérience 2 : Vous pouvez expérimenter une autre procédure d'échange, mais cette fois sans relancer le serveur de noms secondaire. Modifiez d'abord sur les deux serveurs le délai de rafraîchissement et mettez le à 2 ou 3 minute. Relancez les services. Modifiez sur le serveur primaire le N° de série dans l'enregistrement SOA, comme si vous aviez modifié les fichiers de ressources de ns1 et relancez le service. Si vous attendez, vous verrez la synchronisation s'opérer (Trace dans les fichiers de logs). Vous découvrez ainsi le mode de fonctionnement de synchronisation des serveurs de noms sur internet.

Remarque : Si vous voulez, sur ces serveurs assurer la gestion de plusieurs domaines, il suffit de rajouter les définitions de ressources pour ces domaines, puis de déclarer ces zones dans /etc/named.conf.

Notez également que la notion d'autorité est différente de la notion de serveur maître ou serveur esclave. En effet si vous avez en charge la gestion de deux zones (Z1 et Z2), vous pouvez mettre deux serveurs ayant autorité sur ces zones (ns1 et ns2), par contre ns1 peut être serveur maître pour Z1 et secondaire pour Z2, et ns2 peut être serveur maître pour Z2 et esclave pour Z1.

1.10.3. Annexe 3 - Mise en place d'une délégation de zone

Prenons l'exemple d'une zone "sd" d'adresse 192.168.254.0, rattachée à foo.org. Nous allons mettre en place une délégation de zone pour "sd". La résolution des noms de la zone sd.foo.org est prise en charge par les

serveurs de noms de la zone "sd", nous n'avons donc pas à nous en occuper. Par contre nous devons déclarer ces serveurs afin de maintenir la cohérence de la hiérarchie.

Configuration de la délégation : Sur le serveur de noms de la zone foo.org il faut rajouter les enregistrements qui décrivent les serveurs de noms de la zone sd.foo.org dans le fichier de zone db.foo.org.

```
sd 86400 NS ns1.sd.foo.org.
86400 NS ns2.sd.foo.org.
```

Et les enregistrements qui déterminent les adresses de ces serveurs de noms.

```
ns1.sd.foo.org. IN A 192.168.254.1
ns2.sd.foo.org. IN A 192.168.254.2
```

La délégation de la zone in-addr.arpa : Dans la pratique, cette délégation est différente car la zone inverse ne dépend pas de la zone supérieure, mais d'une autre entité (in-addr). Le processus est donc un peu différent.

Pourquoi ? Parce que cette zone reverse est gérée par l'entité qui gère l'espace 192.168.0 à 192.168.255 et il est fort probable que ce n'est pas la zone foo qui assure la résolution inverse pour tous les réseaux compris entre 192.168.0 et 192.168.255.

Ceci dit, cela n'empêche pas de réaliser cela sur une maquette. Il est possible de mettre en place cette résolution inverse. Nous allons donc considérer que la zone foo.org assure la résolution de noms inverse du réseau 192.168.254. Ce reviendrait à considérer que dans la réalité, la zone "sd" serait un sous domaine de "foo". La configuration ici est simple, les masques de sous-réseaux utilisés ici sont ceux par défaut des classes (255.255.255.0) pour la classe C. Le principe pour la zone inverse est identique à celui de la zone directe. Il suffit de rajouter dans le fichier db.0.168.192 les enregistrements suivants :

```
sd.foo.org.                         IN NS    ns1.sd.foo.org.
                                    IN NS    ns2.sd.foo.org.
1.0.168.192.in-addr.arpa   86400  IN  PTR  ns1.sd.foo.org.
2.0.168.192.in-addr.arpa   86400  IN  PTR  ns2.sd.foo.org.
```

2. Installation du service DNS - TD

2.1. Présentation - le contexte

Vous utilisez deux machines M1 et M2.

M1 sera serveur primaire de votre zone, il est également serveur HTTP, serveur FTP, serveur de messagerie et serveur de news.

M2 sera client de M1 pour les trois premières parties du TP et serveur secondaire pour la quatrième partie.

Vous prendrez l’adresse de réseau 192.168.x.0. "x" est variable pour chacun des binômes du groupe. La valeur sera donnée par votre enseignant. Vous remplacerez x par la valeur fournie tout au long de ce document (TD et TP).

Votre domaine est "couleur" ou "couleur" est une variable que vous donnera votre enseignant. Couleur prendra une des valeurs "rouge", "vert", "bleu"....

On considère que M1 est serveur web, serveur ftp, serveur de messagerie et serveur de news.

Voici les noms qui sont assignés :

- Serveur de noms primaire : ns1
- Serveur HTTP : www
- Serveur ftp : ftp
- Serveur de noms secondaire : ns2
- Serveur de mail : mail
- Serveur de news : news

Rédigez les éléments des fichiers named.conf des serveurs primaires et secondaires de votre zone. Vous rédigerez également les fichiers de ressources de la zone "couleur.org".

Vous utiliserez les documents fournis dans la fiche de cours et en annexe.

3. Installation du service DNS - TP

3.1. Présentation

Vous utilisez deux machines M1 et M2. Le TP comporte quatre parties.

1. Première partie : préparation de votre environnement réseau client et serveur
2. Deuxième partie : configuration de la résolution de noms pour la zone directe :

   M1 sera serveur de noms

   M2 sera client de M1

   Test de la configuration à l’aide des commandes ping, et de requêtes ftp, http

3. Troisième partie : configuration de la résolution de noms pour la zone reverse

   Test de la configuration à l'aide de dig

4. Quatrième partie : mise en place du serveur secondaire, modification de l'enregistrement SOA du serveur primaire. :

   Test du transfert de zone

Vous utiliserez les documents réalisés en TD.

3.2. Préparation de votre environnement réseau client et serveur

Ouvrez une session et passez administrateur

Renommez sur les deux machines les fichiers /etc/hosts (mv /etc/hosts /etc/hosts.original) afin d'éviter les effets de bords sur la résolution de noms.

3.3. Installation du serveur de noms primaire

Procédure de configuration du serveur

Vérifiez que vous avez bien les fichiers de configuration de la zone locale, sinon vous devez commencer par là. Vous compléterez ensuite la configuration pour votre zone. Faites une copie de sauvegarde de ces fichiers.

- /etc/bind/named.conf (fichier de configuration du serveur de noms primaire),
- /etc/bind/db.couleur.org qui contiendra la description de la correspondance nom-adresse de toutes les machines de votre zone.
- /var/db.couleur.org.rev qui contiendra la correspondance inverse adresse-nom (pour la résolution inverse de nom in-addr.arpa).

Vérifiez et validez la configuration des clés de rndc.conf et named.conf.

3.3.1. Confi guration du service serveur DNS manuellement

Faites la configuration du serveur (fichiers named.conf, ressources, rndc).

Démarrer le serveur.

Vérifier le bon fonctionnement (traces dans les journaux, processus) de named et rndc.

Corrigez tant qu’il y a des erreurs.

3.3.2. Confi guration du service client manuellement

- Les services clients de M1 et M2 doivent être configurés pour utiliser le service de résolution de noms.
- Modifiez sur les deux machines le fichier "/etc/resolv.conf".
- Relancez le service réseau.
- Testez la configuration
- Vérifiez que la résolution de noms fonctionne sur :

  www.couleur.org

  ftp.couleur.org

  mail.couleur.org ....

- Corrigez tant que cela ne fonctionne pas.
- Vérifiez à ’aide la commande "ping", de requêtes FTP ou HTTP à partir d’un client, que le serveur de noms retourne bien les enregistrements.

  Vérifiez à ’aide la commande "dig", que le serveur répond bien sur différents types de requêtes ("dig any", "dig www", "dig soa".

3.4. Confi guration de la zone reverse

Configurez à l’aide des fichiers fournis en annexe, la zone inverse (reverse). Ceci consiste à rajouter une déclaration dans le fichier "named.conf". Créez le fichier de ressource correspondant.

Relancez le service "named", vérifiez les journaux, corrigez les éventuelles erreurs.

Vérifiez à ’aide de "dig" que les requêtes de type "dig ptr" fonctionnent.

3.5. Installation du serveur de noms secondaire

Sur M2 vérifiez que vous avez bine les fichiers de déclaration pour la zone locale. Ajoutez et déclarez les zones directe et inverses pour votre zone.

Activez le serveur secondaire, vérifiez que le service est actif et vérifiez également dans les journaux qu’il n’y a pas d’erreurs.

Vous devez avoir dans "/var/log/daemon", une trace qui confirme le transfert de zone.

N’allez pas plus loin tant que cela n’est pas en parfait état de fonctionnement.

3.5.1. Procédure de test du serveur secondaire

Arrêtez sur le serveur primaire le service named.

Configurez un client pour qu’il puisse utiliser aussi bien le serveur primaire que le serveur secondaire. Ajouter pour cela un enregistrement de déclaration du serveur secondaire sur le client.

Testez le fonctionnement du serveur secondaire, à partir d’un client, en utilisant des requêtes sur :

www.couleur.org ou ftp.couleur.org.

C’est le serveur secondaire qui doit répondre, le serveur primaire étant inactif.

Vérifier cela avec la commande dig.

3.6. Test de l’enregistrement SOA

Modifiez au minimum le temps de rafraîchissement des enregistrements du serveur Primaire. (2 ou 3 mn). Modifiez également le N° de série. Relancez le serveur primaire et vérifiez dans les logs que le transfert de zone s’effectue bien.

Faites une modification sur votre fichier de ressources db.couleur.org et modifiez le N° de série. Attendez quelques minutes, vous devriez trouver une trace de synchronisation des bases de données des serveurs, sans avoir eu besoin de relancer aucun serveur.

Installation d’un serveur NFS

Partage de ressources disques pour les clients Unix avec Network File System.

1. Résumé

Pourquoi un service NFS alors que celui-ci est très peu utilisé sur les environnements Windows et qu’il n’existe à ma connaissance pas de produits libres client ou serveur pour Windows. Pour deux raisons :

La première est que le service NFS est très largement employé dans les environnement Unix/Linux. Si vous avez des machines sous Linux vous utiliserez NFS. Il est donc nécessaire de connaître les procédures de configuration et d’utilisation de ce service.

La deuxième concerne Windows. Vous aurez sans doute un jour envie ou besoin d’installer le produit Windows Services For Unix (WSFU) de Microsoft. Ce produit disponible déjà sous Windows NT4 Server et mis à jour pour Windows 2000, offre de nombreux outils d’administration de type Unix pour Windows, dont un service NFS.

Nous allons voir, dans un environnement Linux, comment utiliser le service NFS.

2. Installation des produits clients et serveurs

Vous pouvez activer NFS par la commande : /etc/init.d/nfs-kernel-server start. Il vous faudra au préalable avoir défini les ressources à partager (exporter).

Les programmes sur lequel s’appuie le service NFS utilisent les RPC (Remote Procedure Call). Ils s’inscrivent donc auprès du service portmap qui met à jour sa table de service rpc. Voici un extrait de ce que donne la commande rpcinfo -p

```
 program vers proto   port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100003    2   udp   2049  nfs
    100003    3   udp   2049  nfs
    100003    2   tcp   2049  nfs
    100003    3   tcp   2049  nfs
    100021    1   udp  33065  nlockmgr
    100021    3   udp  33065  nlockmgr
```

```
    100021    4   udp  33065  nlockmgr
    100021    1   tcp  38399  nlockmgr
    100021    3   tcp  38399  nlockmgr
    100021    4   tcp  38399  nlockmgr
    100005    1   udp    967  mountd
    100005    1   tcp    970  mountd
    100005    2   udp    967  mountd
    100005    2   tcp    970  mountd
    100005    3   udp    967  mountd
    100005    3   tcp    970  mountd
```

Voici maintenant les processus qui doivent être actifs sur le serveur NFS.

portmap gére le catalogue des programmes RPC,

mountd est chargé des opérations de montage/démontage d’arborescence,

nfsd exécute les primitives d’accès aux fichiers - requêtes émanant des clients.

2.1. Les fichiers de configuration du serveur NFS

/etc/exports décrit ce que le serveur exporte, vers quelles machines le serveur exporte, avec quelles autorisations.

Exemple de fichier /etc/exports :

```
# Ressource  Options   Liste_de_Clients
# Exporte /tmp vers la machine "cli" avec possibilité Read Write (rw)
# rw est l’option par défaut
/tmp  cli(rw)
#Exporte "/tmp" en lecture seule vers toutes les machines du réseau
/tmp  *(ro)
```

Les fichiers de configuration du client NFS :

Il n’y a pas de fichier particulier. Il suffit que les programmes soient installés. Les répertoires exportés par un serveur peuvent être "montés" manuellement ou à la demande. Nous verrons comment configurer un fichier sur le poste client, afin qu’un dossier soit "monté" automatiquement au démarrage du client. Il s’agit dans ce cas d’un service permanent.

2.2. Exemple Unix de montage NFS

Prenons la configuration précédente (fichier /etc/exports ci-dessus)

Le client "cli1" monte (importe) /tmp de ns1 sur le répertoire local /tempo en utilisant la commande suivante

$ mount -t nfs ns1:/tmp /tempo -t indique le type de SGF - arborescence NFS -

Une fois montée, l’accès à la ressource est transparent.

En fin d’utilisation, le client démonte l’arborescence /tmp en utilisant la commande suivante : $ umount /tempo

La table des systèmes de fichiers exportés est localisée dans /etc/fstab

A chaque opération de montage ou démontage, le fichier local /etc/mtab est mis à jour. Il contient la liste des systèmes de fichiers montés (arborescence NFS ou non).

Attention : NFS utilise un cache. Si vous ne voulez pas perdre de données, utiliser une procédure de "démontage" des disques ou alors un "shutdown" du poste client. Dans les autres cas, vous risquez de perdre les informations logées en cache.

2.3. Confi guration du serveur

Vérifiez que le noyau supporte le système de fichiers nfs:

Utilisez la commande "more /proc/filesystems", voici ce que vous pouvez obtenir.

```
nodev   pipefs
        ext2
nodev   ramfs
        msdos
        vfat
        iso9660
nodev   usbfs
nodev   nfs
```

Le système de fichiers nfs doit apparaître. S’il n’apparaît pas, c’est que le système n’est pas compilé avec le support de NFS, ou alors il est compilé pour le charger comme un module. Si c’est le cas, vous pouvez charger le module avec la commande :

```
insmod nfs
```

Le module doit apparaître avec la commande "lsmod", et le fichier "/proc/filesystems" est normalemnt modifié.

2.3.1. Le fi chier /etc/exports

Ce fichier est utilisé par les daemons pour déterminer les volumes qui seront exportés (accessibles), et quels seront les permissions à accorder sur ces volumes. Il existe autant de lignes que de points de montage. La structure d’une ligne est de la forme:

PointDeMontage client1(option) clientn(option)

- PointDeMontage est le volume local à exporter,

- Client1 ... Clientn définissent les ordinateurs qui ont le droit d’accéder au volume exporté,

- Option: définit le type d’accès et les permissions.

Exemple de fichier avec la commande "more /etc/exports"

/tmp *.archinet.edu(rw)

/usr/local/man *.archinet.edu(ro)

Le dossier /tmp est exporté en lecture et écriture pour tous les ordinateurs du domaine archinet.edu. Le dossier /usr/local/man en lecture uniquement.

Voici quelques options de montage, utiliser man exports pour avoir la liste exhaustive:

Secure : requiert une authentification

Insecure : ne requiert pas d’authentification

ro | rw : lecture uniquement ou lecture écriture

Noaccess : permet d’exclure une partie de l’arborescence pour des clients donnés

2.4. Configuration et utilisation du client Unix/Linux

2.4.1. Le fichier /etc/fstab

Ce fichier contient une table des volumes montés sur le système. Il est utilisé par les daemons mount, umount, fsck. Les volumes déclarés sont montés au démarrage du système. Voici un extrait de fichier:

```
/dev/hda1   /    ext2      defaults   1 1
/dev/hda2   swap   swap      defaults   0 0
/dev/fd0    /mnt/floppy  ext2      noauto     0 0
/dev/cdrom   /mnt/cdrom  iso9660   user,noauto,ro 0 0
ns1:/usr/local/man  /doc   nfs       rsize=8192,wsize=8192,timeo=14,intr
```

La dernière ligne indique que le volume /usr/local/man, situé sur le serveur "ns1", et qui contient les pages du manuel est un volume nfs, monté sous le nom de local de /doc.

Ce fichier évite d’avoir à "monter" manuellement des systèmes de fichiers, bien que cela puisse s’avérer parfois nécessaire.

2.4.2. Montage manuel de système de fi chiers

La commande souvent utilisée est de la forme "mount -t TypeDeSGF NomDeMontage VolumeMonté"

Vous pourrez avoir toutes les options avec la commande "man mount" ou une aide plus brève avec "mount --help".

Exemple de montage: "mount -t nfs ns1:/usr/local/man /doc"

Le "mtab" est modifié chaque fois que l’utilisateur "monte" ou "démonte" un système de fichiers. Le système tient à jour une table des volumes montés.

Liste des dossiers montés commande "mount"

La commande "mount" sans paramètres, donne la liste des volumes montés. La commande consulte la table maintenue à jour dans le fichier mtab.

2.4.3. La commande showmount

Cette commande permet d’interroger un hôte distant sur les services NFS qu’il offre, et notamment les volumes qu’il exporte.

Attention : L’accès à la commande "mount" n’est, par défaut, autorisée que pour root.

Il faut rajouter l’option "user" dans le fichier /etc/fstab, afin qu’un autre utilisateur puisse accéder à cette commande.

Exemple: /dev/cdrom /mnt/cdrom iso9660 noauto,ro

Devient /dev/cdrom /mnt/cdrom iso9660 user,noauto,ro

La prise en compte des modifications est dynamique.

2.4.4. Autres commandes d’administration

rpcinfo : (par exemple "rpcinfo -p", consulte le catalogue des applications RPC (nfsd, mountd sont des applicatifs RPC parmi d’autres)

nfsstat : fournit des statistiques d’utilisation de NFS.

3. TP

3.1. Première partie

Vous allez configurer un service de partage de disque pour un client Unix. Vous serez, au cours du TP, serveur pour un autre binôme puis client du serveur d’un autre binôme. Vous allez créer deux répertoires partagés qui seront accessibles par le client :

/tmp sur le serveur sera accessible en lecture/écriture

/usr/share/doc sur le serveur sera accessible en lecture pour le client.

Ces répertoires seront montés respectivement sur les répertoires locaux*/mnt/tempo* et */mnt/doc*

Vous pourrez utiliser les commandes : man exports, man mount, man showmount, man fstab, man rpcinfo

1. Créez sur le serveur le fichier "/etc/exports" et déclarez les fichiers exportés. Voici un extrait de la page de manuel :

   EXEMPLE

   ```
   # fichier /etc/exports d’exemple
   /   maître(rw) confiance(rw,no_root_squash)
   /projects    proj*.local.domain(rw)
   /usr   *.local.domain(ro) @trusted(rw)
   /home/joe   pc001(rw,all_squash,anonuid=150,anongid=100)
   /pub (ro,insecure,all_squash)
   ```

   COMMENTAIRE :

   La première ligne exporte l’ensemble du système de fichiers vers les machines maître et confiance. En plus des droits d’écriture, toute conversion d’UID est abandonnée pour l’hôte confiance.

La deuxième et la troisième ligne montrent des exemples de noms d'hôtes génériques, et de sous-réseaux ('@trusted').

La quatrième ligne montre une entrée pour le client PC/NFS présenté plus haut.

La dernière ligne exporte un répertoire public de FTP, à tous les hôtes dans le monde, en effectuant les requêtes sous le compte anonyme. L'option "insecure" permet l'accès aux clients dont l'implémentation NFS n'utilise pas un port réservé.

Activez les services portmap et nfs. Vérifiez qu'ils sont bien actifs.

Voici un exemple de ce que vous pouvez obtenir avec rpcinfo *-p* :

```
 program no_version protocole  no_port
     100000    2   tcp    111  portmapper
     100000    2   udp    111  portmapper
     100011    1   udp    725  rquotad
     100011    2   udp    725  rquotad
     100003    2   udp   2049  nfs
     100005    1   udp   1026  mountd
     100005    1   tcp   1047  mountd
     100005    2   udp   1026  mountd
     100005    2   tcp   1047  mountd
```

2. Vérifiez sur le serveur les fichiers exportés avec la commande "showmount -e"

   Attention, si vous montez une arborescence sur un répertoire local, et que ce répertoire contenait des fichiers, ces derniers seront masqués le temps du montage.

3. Créez sur le client les points de montage, montez les dossiers exportés du serveur et testez les accès à partir du client.

   La forme standard de la commande mount est : "mount -t type périphérique répertoire" avec :

   Type : Type de sgf (fat, vfat, nfs, ext2, minix....) pour nous c'est nfs

   Périphérique : nom du fichier exporté sous la forme NomServeur:NomDossierExporté

   Répertoire : Nom du répertoire local de montage/

   Le type de fichier que vous montez est de type nfs, vous utiliserez l'exemple de la commande ci-dessous :

   mount -t nfs serveurNFS:/usr/share/doc /mnt/doc

Commentaire : La ligne de commande monte le répertoire exporté "/usr/share/doc" du serveur serveurNFS, sur le répertoire local du client "/mnt/doc".

4. Vérifiez les permissions d’accès lecture et lecture/écriture.
5. À partir du client, créez un fichier sur le fs (file system) accessible en écriture.
6. Ouvrez une autre session sur le serveur dans un autre terminal et essayez de démonter les répertoire montés. Que se passe t-il, pourquoi ?
7. Vérifiez sur le serveur les fichiers exportés avec la commande "showmount -a"
8. Démontez les systèmes de fichiers.

3.2. Deuxième partie

Le fichier "/etc/fstab" permet de déclarer tous les points de montage. Editez et modifiez le fichier sur le client afin d’inclure les systèmes de fichiers nfs exportés par le serveur. Utilisez l’exemple que vous avez dans /etc/fstab.

Rajoutez les lignes nécessaires en vous servant de l’exemple ci-dessous.

serveurNFS:/usr/share/doc /mnt/doc nfs user

Commentaires sur la ligne :

serveurNFS:/usr/share/doc, indique que le dossier "/usr/share/doc" est exporté pas le serveur serveurNFS

/mnt/doc, indique que le dossier distant est monté par défaut sur "/mnt/doc"

nfs, indique qu’il s’agit d’un SGF de type NFS,

user, permet à un utilisateur autre que "root" de monter un répertoire exporté par un serveur.

Pour monter et démonter vous pouvez maintenant utiliser les commandes :

"mount /mnt/doc" et "umount /mnt/doc",

Le système lit le fichier fstab et utilise les paramètres déclarés pour le point de montage dans le fichier fstab.

Vérifiez que les modifications que vous avez apportées dans le fichier fstab fonctionnent.

Supprimez l’option "*user*" sur les lignes que vous avez mises dans le fichier "*fstab*", enregistrer. Essayez ensuite de monter l’arborescence en utilisant un compte autre que "*root*". Que se passe-t-il ?

Restaurez l'environnement.

Installation d’un service de messagerie

Comment installer un serveur SMTP, un client IMAP et un client POP3

1. Le service de messagerie électronique

La messagerie électronique est une application très importante et des plus utiles des réseaux. Plus rapide et moins onéreuse que la plupart des autres moyens de communication (télécopie, téléphone, courrier postal, coursier...) la messagerie électronique est un vecteur de plus en plus important dans la communication aussi bien interne qu’externe. Dans l’univers des réseaux TCP/IP, la messagerie SMTP (Simple Mail Transport Protocol) est de loin la plus utilisée, notamment avec sendmail qui est le standard en matière de serveur SMTP sur les machines Unix.

Le logiciel libre Postfix est un gestionnaire de messagerie simple à configurer et conçu pour une sécurité optimale. De plus il est peu gourmand en ressources système et constitue donc une véritable alternative à Sendmail. Le choix de Postfix est légitime tant pour le traitement de flux importants de messages que pour de petites installations.

L’objectif de ce cours est de préparer l’installation et la mise en exploitation de Postfix en lieu et place de Sendmail.

2. Terminologie

2.1. MHS, MTA, UA, DUA

Le MHS, Message Handler System est le système global de messagerie,

Le MTA, Message Transfert Agent est composé de agents. Un agent de routage (sendmail, MS eXchange...) et un agent de transport (SMTP, UUCP).

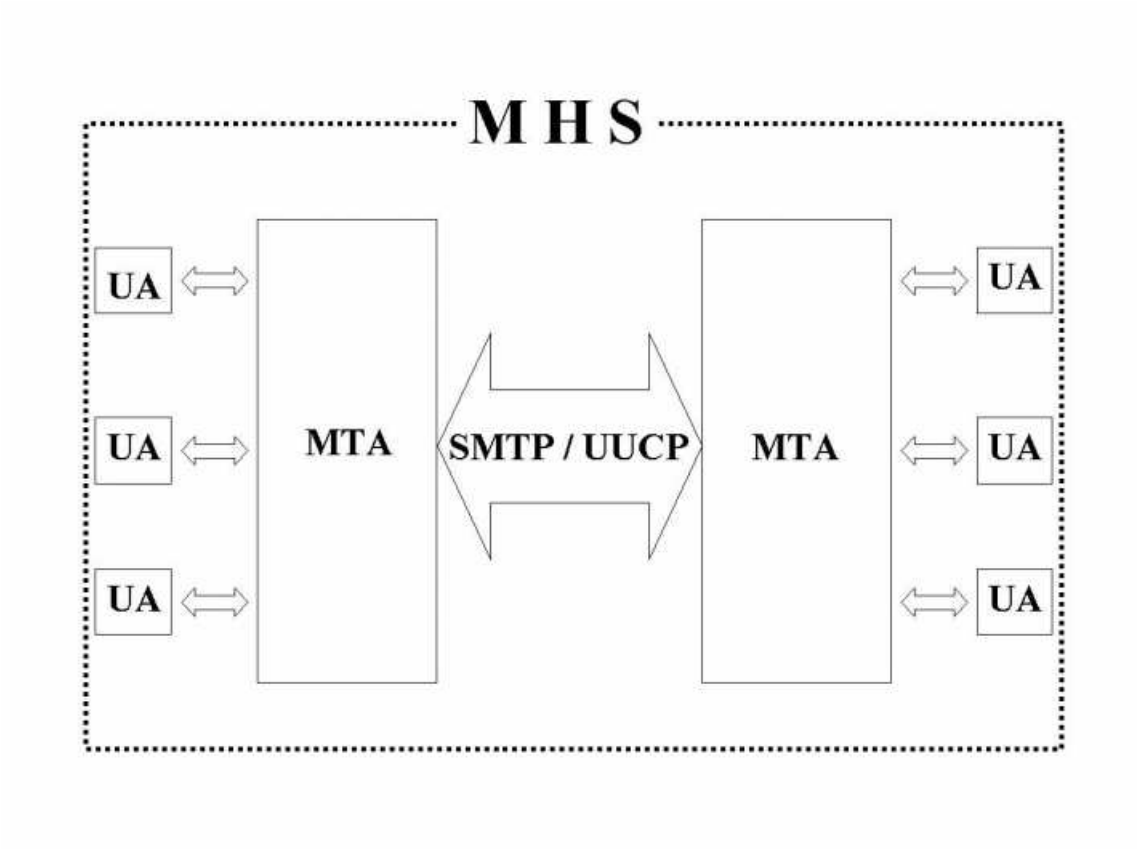
L’agent de routage a pour but d’acheminer le message, en fonction de l’adresse vers son destinataire. Pour nous, avec l’environnement Linux, l’agent de routage est sendmail. L’agent de transport reçoit un message et une direction. Il ne prend aucune décision sur la route à utiliser. Pour nous, protocole de transport peut être SMTP ou UUCP. Le logiciel Sendmail assure les deux fonctions de transport et de routage.

L’UA ou MUA, Message User Agent, est le programme utilisé par le client pour composer, envoyer et recevoir les messages. Pour la composition et l’envoi des messages il existe des programmes comme "mail" sous Linux.

D’autres programmes sont utilisés comme Eudora, Netscape, kmail...On appelle souvent l’UA un "mailer local" si on utilise des outils comme Eudora, Oulook, Mutt, Kmail, ou un "web mail" si on utilise un navigateur comme Mozilla, Netscape ou Internet explorer pour consulter sa messagerie. Ces outils utilisent des protocoles différents. Les protocoles utilisés sont SMTP ou UUCP pour envoyer, et POP3, IMAP, POP3s, IMAPs pour recevoir.

Il existe également un agent (DUA - Delivery User Agent) pour la remise physique du courrier entrant dans la boîte aux lettres de l’utilisateur (BAL). Sur Linux nous utilisons "procmail". Cette remise locale (local delivery) est réalisé par un agent (mail, procmail...) dans des boîtes aux lettres (mailbox) pour mémorisation, (/var/mail/dupont, /var/spool/mail/dupond).

Figure 16. Message Handler System



3. Historique et évolution de sendmail

Sendmail est le routeur de courrier depuis 1982. Il répond aux préconisations de la RFC 822. En 1993, né le standard MIME - RFC 1521 (Multipurpose Internet Mail Extensions), puis en 1994 les extensions du service SMTP (RFC 1652, 1869) pour le transfert caractères 8 bits.

3.1. Mime

Le but de MIME est de standardiser les méthodes de transfert de données 8 bits,structurer le corps du message en contenus (body-parts),standardiser les différents contenus possibles. Un en-tête est rajouté à ceux définis dans le RFC 822 : Mime-version:1.0

3.1.1. Le standard MIME

MIME supporte plusieurs type d'encodage comme :

1. Texte 7 bits, US-ASCII
2. Quoted-Printable (Caractère non US-ASCII remplacé par une séquence =XY, XY étant le code hexadécimal du caractère.)
3. Base 64 (Texte, image, son)
4. 8Bits (les lignes sont composées de caractères 8 bits, il faut préciser l'alphabet : iso-latin1)
5. Binary

La strucure d'un message MIME est standardisée par des entêtes supplémentaires qui décrivent la structure et le type de contenu (format des données) du message.

Exemple de déclaration décrivant la structure :

1. Multipart/mixed
2. Multipart/parallel (plusieurs parties avec affichage en parallèle.)
3. Multipart/digest (d'autre(s) message(s) inclus dans le message)
4. Multipart/alternative (partie du message affichée suivant l'environnement du correspondant.)

Exemple de déclaration décrivant le format des données

1. Text/plain : charset=iso-8859-1
2. Text/richtext
3. Image/gif
4. Image/jpeg
5. Audio/basic
6. Video/mpeg
7. Application/octet-stream : exemple word
8. Application/postscript

3.1.2. Exemple de message

```
From mascret  Mon Mar 19 08:02:46 2001
Return-Path: <Marcel.Giry@unilim.fr>
Delivered-To: alix.mascret@beaupeyrat.com
```

```
Received: from limdns2.unilim.fr (limdns2.unilim.fr [164.81.1.5])
        by pegase.beaupeyrat.com (Postfix) with ESMTP id AC04237B05
        for <salvaco@beaupeyrat.com>; Mon, 19 Mar 2001 08:02:44 +0100 (CET)
Received: from pctest (modem8.unilim.fr [164.81.1.208])
          by limdns2.unilim.fr (8.9.1a/jtpda-5.3.2) with ESMTP id IAA04253
          ; Mon, 19 Mar 2001 08:02:39 +0100
Message-Id: <4.2.0.58.20010319080303.00950a70@pop.unilim.fr>
X-Sender: xalan@pop.unilim.fr (Unverified)
X-Mailer: QUALCOMM Windows Eudora Pro Version 4.2.0.58
Date: Mon, 19 Mar 2001 08:05:13 +0100
To: salvaco@beaupeyrat.com,
 xalan@univlim.fr
From: Ximian Alan <xalan@univlim.fr>
Subject: Controle IUT2
Mime-Version: 1.0
Content-Type: multipart/mixed;
  boundary="====================_811307==_"
Status: RO
X-Status: A
```

3.1.3. L'importance d'un bon UA

MIME permet l'utilisation de plusieurs types de données (text, audion compressés...) et plusieurs format (rtf, doc, gz, zip...). Il est important de posséder un UA de bonne qualité.

1. Reconnaître et afficher du texte US-ASCII,
2. Reconnaître les autres jeux de caractères et permettre de sauvegarder les contenus non reconnus dans un fichier pour traitement ultérieur
3. Reconnaître et afficher les contenus de type message/RFC822
4. Reconnaître le type Multipart/mixed
5. Reconnaître le type Multipart/alternative
6. Traiter les Multipart non reconnus comme Multipart/mixed
7. Décoder les contenus de Application/* si l'encodage quoted-printable ou base64 est utilisé, puis offrir de sauver le résultat dans un fichier.

4. Pourquoi Postfix

Le serveur de messagerie standard sur les systèmes Unix est le serveur Sendmail. Sendmail a fait ses preuves. L'inconvénient est son mode de configuration. Toutes les fonctions de messagerie sont réalisées par un seul programme. Sa structure est dite monolithique et la configuration (fichier sendmail.cf) en est d'autant plus compliquée. Ce phénomène s'accroit avec l'amplification de l'utilisation du service de messagerie (augmentation de fréquence/volume) et avec l'exposition aux tentatives de piratage des serveurs de messagerie. Il existe d'autres serveurs de messagerie sur Unix (QMail, Z-mailer...) tous présentent des inconvénients au niveau utilisation de la bande passante, inter-opérabilté, respect des RFC, facilité de configuration, sécurité...

L'objectif de postfix est d'apporter une solution à ces différents problèmes.

4.1. Buts premiers : un nouveau MTA sous Unix

1. bénéficier de l'expérience de sendmail
2. facile à administrer : ce qui est facile à comprendre est plus facile à sécuriser.
3. rapide et évolutif : le trafic SMTP de 1999 n'est pas celui de 1980. Il faut pouvoir faire un logiciel supportant les sites énormes (ISP, Accès des grosses entreprises, ...)
4. compatibilité sendmail maximale

Il assure égalemetn un compatibilité et le support :

1. des MUA existants (pine,mutt,mail, ...)
2. des gestionnaires de liste (majordomo, sympa, ...)
3. des formats de boîte aux lettres (mh, mbox, qmail-dir, ...)
4. des agents d'acheminement local (procmail, deliver, cyrus, ...)
5. des configurations (UUCP, réécriture, mailertable, ...)
6. des utilisateurs (alias, .forward, ...)
7. des RFCs

4.2. l'Auteur

L'Auteur - Wietse Venema - est connu pour ses contributions à la sécurité et aux logiciels libres. Il est également auteur de TCP_Wrapper et d'un portmap sécurisé. Il est co-auteur avec D. Farmer de SATAN. Il travaille au Watson Research Center d'IBM.

Postfix est un logiciel libre. Le site officiel est www.postfix.org.

5. Architecture de postfi x

Postfix (voir bigpicture.gif) est architecturé autour d'un module de réception des messages (voir inbound.gif) et de celui qui permet de délivrer ces messages (voir outbound.gif).

Figure 17. Architecture de Postfix

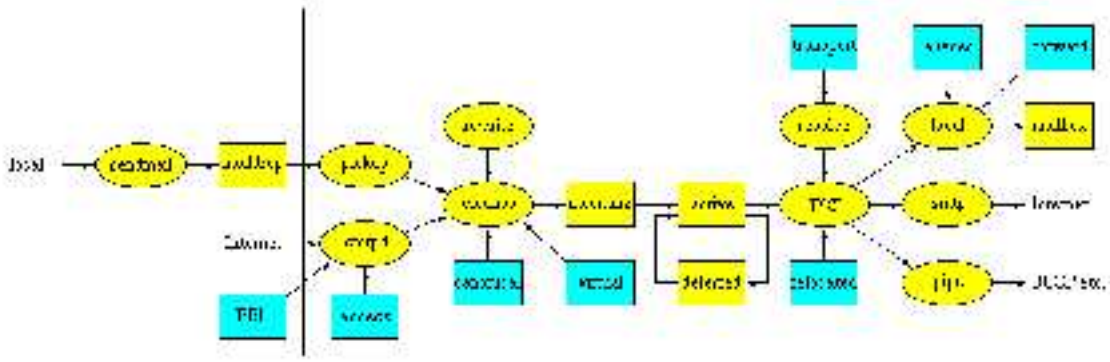


Figure 18. Réception des messages

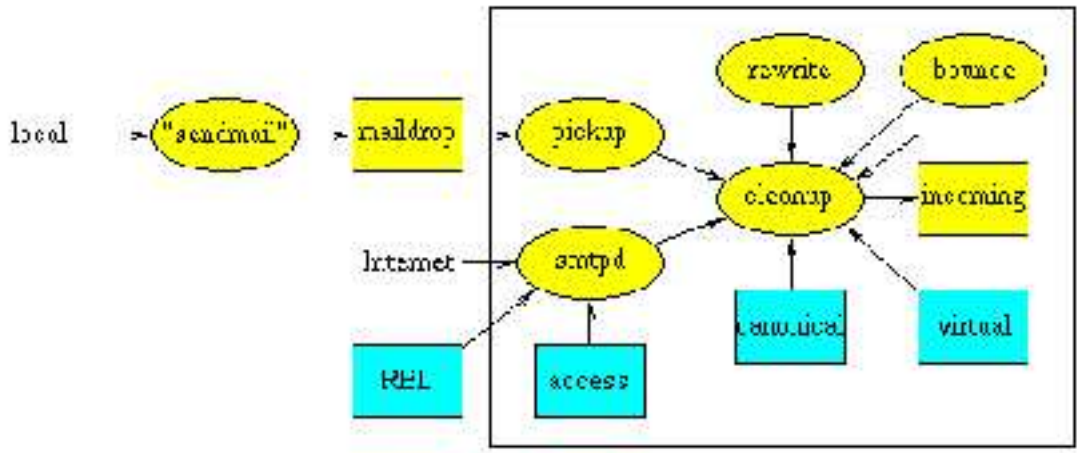
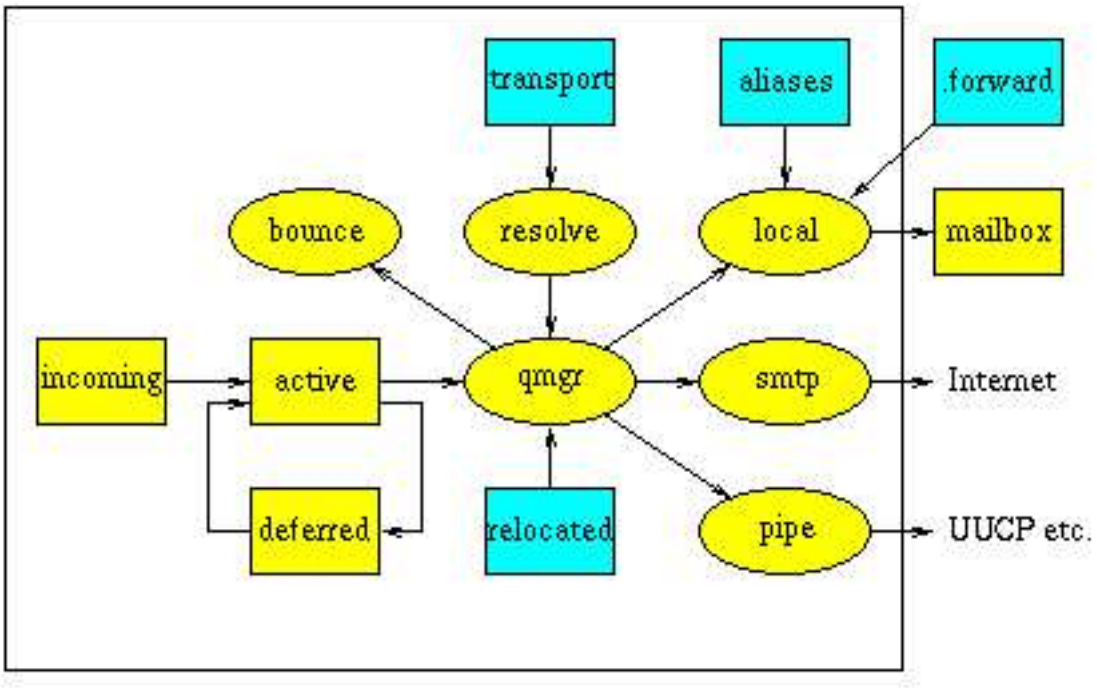


Figure 19. Traitement des messages



5.1. La réception des messages (entrées)

Quand un message doit être traité par un système Postfix, le passage obligé est la file "incoming".

Si le message est posté localement, il est déposé dans un répertoire en accès "écriture possible pour tout le monde". Le démon "pickup" le traitera à partir de là. Ce démon procède à une première phase d’analyse des couriers (headers) afin de protéger le reste du système.

Si le message provient d’un réseau, le message est traité par un serveur SMTP. Certaines règles de sécurité et de contrôles sont déjà effectuées.

Les messages peuvent être générés par Postfix lui même ou par un robot afin de prévénir l’administrateur des erreurs, adresses introuvables, tentatives de violations des règles, problèmes de protocoles...

Les messages peuvent être redistribués par des entrées dans les fichiers d’alias ou des fichiers ".forward".

Le démon "cleanup" représente la période finale de traitement d’un message, notamment la vérification de l’entête du message (complétude user@fqdn), la réécriture d’adresse, le dépôt du message dans la file incoming, l’avertissement du gestionnaire de liste.

5.2. Délivrer les messages

Quand un message est arrivé dans la file "incoming", l’étape suivante consiste à le délivrer. Ceci est pris en charge par le gestionnaire de file qui est le coeur du système de Postfix. Il contacte un agent (local, smtp, lmtp, pipe) chargé de délivrer les messages en lui communiquant des paramètres (localisation du message, nom/adresse de l’emetteur, nom(s)/adresse(s) du/des destinataires, machine hôte de destination....

Le gestionnaire de liste maintient une liste séparée pour les courriers ne pouvant être délivrés immédiatement (deferred).

Les messages ne pouvant être définitivement délivrés (bounces) génèrent une trace d’information dans les journaux.

Sur Linux, l’agent de traitement local des messages est le plus souvent "procmail". Il doit pouvoir traiter des structures de boîtes aux lettres conformes au standard Unix, utiliser les aliases, les redirections ".forward"...

L’agent de traitement pour l’acheminement distant des messages s’appuie sur le protocole SMTP et utilise le port 25.

Les différents démons sont activés "à la demande" par un super serveur (master daemon) un peu à la façon d’inetd.

5.3. Une fonction / un programme

Chaque grande fonction de postfix est prise en charge par un programme indépendant.

1. Lecture des messages locaux
2. Réception SMTP
3. Réécriture d’adresse
4. Envoi SMTP
5. Délivrance locale
6. Traitement des erreurs (bounces)
7. Gestion des files

5.4. Apports en termes de sécurité:

Cette option présente plusieurs avantages.

1. Décomposition = programmes plus petits et plus lisibles
2. Plus difficile à casser ou circonvenir
3. Chroot plus facile
4. Les programmes ne se font pas confiance : isolation de chaque fonction

5.5. Communication interprocessus par sockets Unix ou file (FIFO)

1. Portabilité aisée
2. Messages courts dans les sockets
3. Ne pas faire confiance aux données

5.6. Semi résidence

1. Les démons sont réutilisés et contrôlés par un super démon "master" qui les crée à la demande.
2. Nombre maximum pour chaque fonction : contrôle précis du fonctionnement, sécurité contre le "déni de service" (DOS)
3. Temps d’inactivité paramétrable

5.7. Files d’attente multiples

1. maildrop : messages locaux postés par sendmail
2. incoming : messages en cours de réécriture et de nettoyage
3. active : messages en cours ou en attente de transport
4. deferred : messages en attente
5. defer : arborescence d’attente (hachée pour éviter les trop gros répertoires -- problème dans Sendmail)

6. Configuration et fichiers de configuration de Postfix

Les outils d’administration et de maintenance sont dans /usr/sbin. Voici les principaux.

1. postalias sert à maintenir la base de données des alias
2. newaliases (/usr/bin) assure la compatibilité avec sendmail pour la base de donnes des alias
3. postcat affiche le contenu des files d’attentes.
4. postconf affiche les paramètres de Postfix contenus dans fichier "main.cf"
5. postlog, sert à gérer les logs (réalisation de scripts)
6. postqueue, permet de gérer et administrer les files d’attentes.

Les journaux (logs) sont dans /var/log

6.1. Configuration - Extrait du fichier /etc/postfix/master.cf

Il définit les démons à lancer, leur nombre et les "transports"

```
# ==========================================================================
```

```
# service type  private unpriv  chroot  wakeup  maxproc command args
#               (yes)   (yes)   (yes)   (never) (50)
# ==========================================================================
smtp      inet  n       -       y       -       -       smtpd
smtps     inet  n       -       y       -       -       smtpd \
           -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
submission inet n       -       y       -       -       smtpd \
           -o smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=yes
pickup    fifo  n       n       y       60      1       pickup
cleanup   unix  -       -       y       -       0       cleanup
qmgr      fifo  n       -       y       300     1       qmgr
#qmgr     fifo  n       -       y       300     1       nqmgr
tlsmgr    fifo  -       -       y       300     1       tlsmgr
rewrite   unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
flush     unix  -       -       y       1000?   0       flush
smtp      unix  -       -       y       -       -       smtp
showq     unix  n       -       y       -       -       showq
error     unix  -       -       y       -       -       error
local     unix  -       n       n       -       -       local
virtual   unix  -       n       n       -       -       virtual
lmtp      unix  -       -       n       -       -       lmtp
[...]
```

6.2. Le fichier de configuration /etc/postfix/main.cf

Si postfix n’a pas été préalablement configuré, vous n’avez pas de fichier de configuration main.cf. Vous pouvez utiliser la commande :

```
dpkg-reconfigure postfix
```

Utilisez les paramètres suivants pour une configuration minimale :

```
#$NOM_MACHINE est le nom d’hôte de votre machine
local only
$NOM_MACHINE
Append Domain no
Destination $NOM_MACHINE
Local Network 127.0.0.0/8
Use Procmail Yes
Siez Mail Box 0
Char Def Local Adress +
```

Le fichier "main.cf" contient tous les paramètres de postfix. Ceux-ci peuvent être affichés avec la commande postconf.

Voici un exmple de main.cf que vous pourrez réutiliser pour les TP

```
# Vous avez un fichier complet et commenté
# /usr/share/postfix/main.cf.dist
```

```
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
program_directory = /usr/lib/postfix

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
setgid_group = postdrop
biff = no
2bounce_notice_recipient = postmaster

# appending .domain is the MUA’s job.
append_dot_mydomain = no
myhostname = NomHote.foo.org
mydomain = foo.org
mydestination = $myhostname, localhost.$mydomain $mydomain
myhostname = NomHote.foo.org
myorigin = $mydomain
myorigin = /etc/mailname

alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
# /etc/mailname contient l’équivalent de $MYHOSTNAME

mynetworks = 127.0.0.0/8 192.168.0.0/24
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
relay_domains = $mydestination
relayhost = $mydomain
smtpd_recipient_restrictions = permit_mynetworks,check_relay_domains
```

Pour une configuration initiale remplir myhostname, mydomain, myorigin, mydestination, relayhost.

6.3. Le fichier de configuration des aliases /etc/aliases

Il sert à la création des aliases, par exemple : jean.dudognon sera l’alias du compte système jddgn. Le courrier sera adressé à jean.dudognon@domaine.dom, mais sera délivré dans la boîte du compte jddgn, c’est à dire physiquement dans "/var/spool/mail/jddgn".

Le fichier /etc/postfix/aliases est de type texte. C’est celui-ci que vous modifiez. Après chaque modification du fichier source utiliser la commande "newaliases" ou "postaliases hash:/etc/postfix/aliases" qui met à jour le fichier de bases de données "/etc/postfix/aliases.db".

6.4. Surveillance et maintenance de postfix

La maintenance est réalisée à l’aide des commandes externes. Autrement les transactions sont journalisées par le démon syslogd.

```
Oct 31 11:23:26 uranus postfix/master[2745]: daemon started

    uranus postfix/smtpd[2753]: connect from unknown[192.168.1.1]
```

```
 uranus postfix/smtpd[2753]: 82BF05769B: client=unknown[192.168.1.1]
 uranus postfix/cleanup[2754]: 82BF05769B:
message-id=<20011031102453.82BF05769B@uranus.foo.org>
 uranus postfix/qmgr[2749]: 82BF05769B:
        from=<mlx@foo.org>, size=318, nrcpt=1 (queue active)
 uranus postfix/local[2756]: 82BF05769B:
to=<mlx@foo.org>, relay=local, delay=94,
        status=sent ("|/usr/bin/procmail /etc/procmail.rc")
 uranus postfix/smtpd[2753]: disconnect from unknown[192.168.1.1]
```

7. Structure des messages

Un messages est schématiquement composé de deux parties, une entête et un corps. Ces deux parties sont séparées par une ligne blanche.

```
L’entête est découpée ainsi :
 FROM:     expéditeur
 TO:     destinaire(s)
 CC:     copie à
 BCC:  copie  aveugle
 REPLY-TO:     adresse de réponse
 ERROR-TO:     adresse en cas d ’erreurs
 DATE:     date expédition
 RECEIVED     informations de transferts
 MESSAGE-ID:   identificateur unique de msg
 SUBJECT:     sujet
```

8. Le dialogue entre le client et le serveur

Le dialogue est défini par le protocole SMTP selon un schéma client/serveur. Sur le client, un démon (programme sendmail ou smtpd par exemple) attend les requêtes TCP sur le port 25 d’un client (le programme mail par exemple).Le dialogue est en ASCII. Pour tester utilisez la commande :

"telnet Serveur_SMTP 25" ou encore "sendmail -v -bs"

Exemple de dialogue : La chaîne ">>>" n’apparaît pas, c’est juste pour distinguer les commandes client.

```
[mlx@uranus mlx]$ telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is ’^]’.
220 uranus.foo.org ESMTP Postfix
>>> EHLO uranus
250-uranus.foo.org
250-PIPELINING
250-SIZE 10240000
250-ETRN
```

```
250 8BITMIME
>>> MAIL FROM:<mlx@uranus.foo.org>
250 Ok
>>> RCPT TO:<mlx@foo.org>
250 Ok
>>>DATA
354 End data with <CR><LF>.<CR><LF>
Message de test
.
250 Ok: queued as C21B15769B
>>> QUIT
221 Bye
Connection closed by foreign host.
You have new mail in /var/spool/mail/mlx
```

9. PostOFFICE

Le service POP - Postoffice Protocole est utilisé par les logiciels clients (netscape, Eudora, Outlook...) pour relever le courrier sur les serveurs de messagerie. Le client pop utilise un couple Nom d’utilisateur/mot de passe pour la phase identification/authentification par le serveur. Le service pop3d reste en écoute sur le port 110. Il est généralemnt lancé par le démon inetd ou xinetd. Pour activer le service pop3 il suffit de décommenter la ligne correspondante dans le fichier /etc/inetd.conf. Voici des exemples de lignes que vous pouvez avoir dans votre fichier inetd.conf :

```
#:MAIL: Mail, news and uucp services.
imap2   stream  tcp     nowait  root   /usr/sbin/tcpd /usr/sbin/imapd
imaps   stream  tcp     nowait  root   /usr/sbin/tcpd /usr/sbin/imapd
pop3    stream  tcp     nowait  root   /usr/sbin/tcpd /usr/sbin/ipop3d
pop3s   stream  tcp     nowait  root   /usr/sbin/tcpd /usr/sbin/ipop3d
#imaps   stream  tcp nowait  root    /usr/sbin/tcpd  \
                  /usr/sbin/sslwrap  -nocert -addr 127.0.0.1 -port 143
#pop3s   stream  tcp nowait  root    /usr/sbin/tcpd  \
                  /usr/sbin/sslwrap  -nocert -addr 127.0.0.1 -port 110
```

ou d’activer le service " disable = no" dans "/etc/xinetd.d/pop3", si vous utilisez xinetd.d.

10. IMAP (Internet Message Access Protocol)

Pop a été conçu pour la consultation "hors ligne", (off line). Imap permet la consultation hors ligne, mais également "en ligne", selon un processus interactif entre le client et le serveur. Les messages ne sont plus rapatriés sur le client. Ils restent en dépôt sur le serveur jusqu’à ce que l’utilisateur demande explicitement la suppression ou le transfert.

Ce procédé est particulièrement interessant pour les utilisateurs mobiles. Ils peuvent consulter leur messages à partir de machines ou de lieux non définis à l’avance.

Comme la connexion au serveur est permanente pendant la durée du traitement, il présente l’inconvénient d’un surcoût financier du à la liaison téléphonique.

Ces services utilisent des protocoles/ports différents. Ils peuvent cohabiter simultanément sur le même serveur. Un utilisateur peut utiliser selon ses besoins l’un ou l’autre des services POP ou IMAP.

Vous devrez installer et configurer sur les postes clients, un client IMAP (Netscape messenger, Kmail...).

Des logiciels d’interface sur le serveur comme IMP (www.imp.org), permettent de transformer le serveur IMAP en serveur "webmail". Les clients pourront alors utiliser n’importe quel navigateur pour consulter leur boîte aux lettres. Le CRU, (Comité Réseau des université - www.cru.fr), a fait une étude sur les principaux produits qui pouvaient être utilisés.

11. Remarques sur pop3 et imap

Les ports utilisés par les services pop3, pop3s, imap, impas, sont déclarés dans le fichiers etc/services. Voici ce que donne le lancement d’inetd :

```
root@knoppix:/home/knoppix# netstat -atup | grep LISTEN
tcp        0      0 *:imaps     *:*       LISTEN      376/inetd
tcp        0      0 *:pop3s     *:*       LISTEN      376/inetd
tcp        0      0 *:pop3      *:*       LISTEN      376/inetd
tcp        0      0 *:imap2     *:*       LISTEN      376/inetd
```

Dans un soucis de sécrurité, les applications récentes ne supportent plus les transactions "en clair" sur le réseau. Cela signifie que les applications sont compilées pour utiliser les protocoles d’encryptage TLS/SSL. Vous devrez en tenir compte dans la configuration de vos clients et vous utiliserez pop3s et imaps.

Pour en savoir plus, vous pouvez consulter :

```
/usr/share/doc/ipopd/README.Debian
/usr/share/doc/libc-client2003debian/README.Debian
/usr/share/doc/libc-client2003debian/md5.
/usr/share/doc/libc-client2002/md5.txt
/usr/share/doc/libc-client2003debian/imaprc.txt
```

Configuration d’un système de messagerie

Comment installer un serveur SMTP, un client IMAP et un client POP3

1. TP - Installation de postfix

Vous allez installer successivement :

1. le serveur de messagerie postfix puis tester son fonctionnement,
2. un serveur de remise de courrier pop3 et imap
3. un client pop3 et imap puis tester son fonctionnement,

Pour imap on utilisera uw-imapd/uw-imapd-ssl, pour pop3, on utilisera ipopd. Pour les préconfigurer vous utiliserez les commandes :

```
dpkg-reconfigure uw-imapd
dpkg-reconfigure ipopd
```

Vous sélectionnerez pop3 et pop3/ssl, imap4 et imap/ssl. Attention pour imap4, c’est imap2 qu’il faut sélectionner. C’est bizarre mais c’est comme ça ;-) imap3 est devenu obsolète.

La procédure de configuration, génère des certificats dans /etc/ssl/certs/

2. DNS - Préparation préalable

Vous allez déjà préparer votre serveur de nom. Le serveur de nom primaire sera également serveur SMTP (enregistrement MX - Mail eXchanger). Si votre serveur de nom s’appelle "ns1", rajouter les enregistrements suivants dans le fichier de configuration de votre zone :

```
# On définit la machine qui achemine le courrier pour
# user@ns1.VotreDomaine.Dom
@ IN MX 10 ns1.VotreDomaine.Dom
#On définit un alias pour le courrier envoyé à
mail IN CNAME ns1
#On définit un alias pour le courrier envoyé à partir de
```

```
smtp IN CNAME ns1
#On définit un alias pour le serveur pop et pour imap
pop IN CNAME ns1
imap IN CNAME ns1
```

Relancer le service dns. Les commandes suivantes doivent fonctionner à partir d'un client du domaine :

```
ping ns1.VotreDomaine.Dom
ping smtp.VotreDomaine.Dom
ping mail.VotreDomaine.Dom
ping pop.VotreDomaine.Dom
ping imap.VotreDomaine.Dom
```

3. Configuration du serveur postfix.

Vous allez successivement configurer un serveur SMTP Postfix, tester la configuration, installer les serveur pop et imap, tester le fonctionnemnt de l'ensemble.

3.1. Installation du serveur SMTP

Configurer votre machine pour un service minimum (pas de liste, pas de réécriture d'adresse...).

Utilisez l'exemple de configuration de main.cf et la liste des variables à configurer donnés dans la fiche de cours, afin de mettre en place un service minimum.

Activez le service avec la commande "/etc/init.d/postfix start". Vérifiez le bon démarrage du serveur dans le fichier de log et dans la table des processus (ps axf). Vous devriez obtenir quelque chose comme :

```
 2745 ?        S      0:00 /usr/lib/postfix/master
 2748 ?        S      0:00  \_ pickup -l -t fifo -c
 2749 ?        S      0:00  \_ qmgr -l -t fifo -u -c
 2750 ?        S      0:00  \_ tlsmgr -l -t fifo -u -c
```

Vérifiez également les traces dans le fichier de journalisation.

3.2. Test de la configuration du serveur SMTP

Créez sur la machine locale deux comptes systèmes pour les tests. cpt1 et cpt2 par exemple.

Ouvrez une session sous le compte cpt1 afin de réaliser un envoi de mail pour cpt2.

Lancez une transaction "telnet VotreServeur 25", et réalisez un dialogue similaire à celui décrit en TD Le message doit être délivré dans la boîte de cpt2. Utilisez la commande "ps axf" pour voir le chargement des différents démons.

Rélisez l’opération à l’aide du programme "mail". Vérifiez que le message est bien délivré.

Avant de terminer la transaction, identifiez la session avec la commande netstat :

```
netstat -atup | grep ESTABLISHED
```

3.3. Installation du serveur PostOFFICE Pop3

La configuration du service pop est des plus simple. Il est même possible qu’il soit déjà actif. Décommentez la ligne dans le fichier "/etc/inetd.conf" ou utilisez dpkg-reconfigure.

Vérifier le fichier /etc/inetd.conf, relancez au besoin le service inetd.

Pop and imap mail services

Avec xinet, la configuration est dans /etc/xinetd.d. Editez les fichiers correspondant aux différents services. Par exemple le fichier /etc/xinetd.d/pop3s

```
# default: off
# The POP3S service allows remote users to access their mail \
# using an POP3 client with SSL support such as fetchmail.

service pop3s
{
socket_type = stream
wait = no
user = root
server = /usr/sbin/ipop3d
log_on_success += USERID
log_on_failure += USERID
disable = no
}
```

Dans inetd.conf décommentez la ligne. Dans xinetd, mettez la variable "disable à no".

Relancer le service inetd ou xinetd, vérifiez l’ouverture des ports avec la commande netstat.

Identifiez les numéros de ports des services dans le fichier /etc/services.

3.4. Test du serveur Pop3

Vous allez réaliser l’opération à partir de la machine locale et d’une machine distante. La résolution de nom doit fonctionner, sinon utilisez les adresses IP. Vous utiliserez kmail ou le client de messagerie de Mozilla.

1. Sur la machine locale qui est votre serveur SMTP et serveur POP3, configurez le client de messagerie avec les paramètres suivants :

   ```
   Serveur smtp : Nom de votre serveur
   Serveur POP :  Nom de votre serveur POP
   Votre compte d’utilisateur
   Votre mot de passe
   ```

   Testez l’envoi et la réception de message.

   Renseignez bien le numéro de port. Dans kmail, l’onglet "extras" vous donne accès à un bouton "tester ce que le serveur peut gérer", et va vous renseigner sur le support de ssl ou tls du serveur.

   Avec un client pop, les messages sont, par défaut, téléchargés depuis le serveur sur le client. La procédure supprime les fichiers téléchargés sur le serveur. Cette option est configurable sur la majorité des clients.

   Vérifiez que les fichiers sont bien supprimés sur le serveur.

   Réïtérez l’envoi de message en mettant un fichier attaché (par exemple un fichier xls). Vérifier et relevez la description MIME du message.

2. Sur un client configurez Nestcape Messenger avec les paramètres suivants :

   ```
   Serveur smtp : Nom de votre serveur (Machine distante)
   Serveur POP :  Nom de votre serveur POP3 (Machine distante)
   Votre compte d’utilisateur
   Votre mot de passe
   ```

   Testez l’envoi et la réception de message.

Identifiez les transactions dans le fichier de log.

3.5. Utilisation des alias

Créez un compte utilisateur pn,

Créez un alias prenom.nom pour ce compte système dans le fichier /etc/aliases.

Mettez à jour le fichier "/etc/aliases/db".

Vérifiez que les messages envoyés à :

pn@votredomaine ou prenom.nom@votredomaine doivent tous être correctement délivrés.

3.6. Utilisation des listes

Ouvrez à l’aide d’un éditeur le fichier /etc/aliases

Créez une liste de la façon suivante :

```
maliste: cpt1, cpt2
```

Enregistrez et régénérez le fichier aliases.db

Envoyez un message à maliste@foo.org, vérifiez que tous les membres de la liste ont bien reçu le message.

3.7. La gestion des erreurs

Certaines erreurs "systèmes" sont gérés par un compte particulier "MAILER_DAEMON". Ce compte est en général un alias vers "postmaster", qui, lui même redirige sur le compte de l’administrateur en fonction. Il agit commem un "robot" notamment quand un message ne peut être délivré.

Procédure de test

Envoyez un message à QuiNexistePas@foo.org

Relevez vos nouveau messages. Normalement, vous êtes averti que votre messgae n’a pas pu être délivré.

3.8. Mise en place du service IMAP sur le serveur

La mise en oeuvre est identique à celle du service Pop3. Configurer le fichier inetd.conf ou le fichier /etc/xinetd.d/imap. Relancer le service serveur (inetd ou xinetd).

Vous allez tester le bon fonctionnement de votre serveur : La commande "ps aux | grep imap", ne donne rien car le serveur imap est lancé ’à la demande’ par le serveur inetd ou xinetd.

Par contre la commande "netstat -a | grep LISTEN | grep imap" montre bien qu’un port est bien ouvert en "écoute".

```
tcp        0      0 *:imap                  *:*               LISTEN
```

Tapez la commande "telnet @DeVotreServeurImap 143" pour activer le service imap. Le serveur doit répondre :

```
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
* OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=LOGIN]\
   uranus.foo.org IMAP4rev1 2000.287rh at Sat, 17 Nov 2001 14:14:42 +0100 (CET)
```

Dans une autre sessiosn xterm, la commande "ps aux | grep imap" montre mainetant que le service est maintenant bien dans la liste des processus:

```
 root     11551  0.0  1.1  3664 1448 ?        S    14:14   0:00 imapd
```

et la commande "ps axf"

```
  231 ?        S      0:00 /usr/sbin/inetd
  315 ?        S      0:00  \_ imapd
```

montre bien que le processus imapd dépend (est fils de) inetd.

3.9. Plus loin dans le décryptage

La commande "netstat a | grep imap" donne l’état d’une connexion établie entre un client et le serveur.Socket client TCPsur le port 1024

```
tcp        0      0 *:imaps           *:*               LISTEN
tcp        0      0 *:imap2           *:*               LISTEN
tcp        0      0 knoppix:imap2     knoppix:1025      ESTABLISHED
tcp        0      0 knoppix:1025      knoppix:imap2     ESTABLISHED
```

La commande "fuser 1025/tcp" utilise le pseudo-système de fichiers d’informations sur les processus "/proc" pour identifier "QUI" utilise la connexion tcp sur le port 1025.

```
root@knoppix:/home/knoppix# fuser 1025/tcp
1025/tcp:             364
```

La commande "ls -l /proc/364" donne les indications sur le programme qui utilise cette connexion et montre que c’est une commande "telnet" qui a déclenché le processus.

```
root@knoppix:/home/knoppix# ls -al /proc/364
total 0
dr-xr-xr-x 3  knoppix  knoppix  0 2003-04-16 15:06 .
dr-xr-xr-x 49 root     root     0 2003-04-16 16:52 ..
```

```
-r--r--r-- 1  knoppix  knoppix  0 2003-04-16 15:06 cmdline
-r--r--r-- 1  knoppix  knoppix  0 2003-04-16 15:06 cpu
lrwxrwxrwx 1 knoppix  knoppix 0 2003-04-16 15:06 cwd -> /home/knoppix
-r-------- 1 knoppix  knoppix 0 2003-04-16 15:06 environ
lrwxrwxrwx 1 knoppix  knoppix 0 2003-04-16 15:06 \
                                          exe -> /usr/bin/telnet-ssl
dr-x------ 2 knoppix  knoppix 0 2003-04-16 15:06 fd
-r--r--r-- 1 knoppix  knoppix 0 2003-04-16 15:06 maps
-rw------- 1 knoppix  knoppix 0 2003-04-16 15:06 mem
-r--r--r-- 1 knoppix  knoppix 0 2003-04-16 15:06 mounts
lrwxrwxrwx 1 knoppix  knoppix 0 2003-04-16 15:06 root -> /
-r--r--r-- 1 knoppix  knoppix 0 2003-04-16 15:06 stat
-r--r--r-- 1 knoppix  knoppix 0 2003-04-16 15:06 statm
-r--r--r-- 1 knoppix  knoppix 0 2003-04-16 15:06 status
```

et voir la commande qui a activé cette connection " more /proc/364/cmdline" qui retourne "telnet localhost 143".

3.10. Mise en place du client IMAP

Utilisez Mail & NewsGroup de Mozilla ou kmail par exemple. Dans Mail & News Group, Allez dans le me nu de configuration (Édit) et ajoutez un compte. Prenez un compte imap.

Complétez la configuration de votre client de messagerie

Ouvrez l’application "Messenger", testez l’utilisation du client IMAP.

3.11. Le relayage

Utiliser le "relayage" consiste pour un client A, à utiliser le service serveur SMTP d’un domaine B pour inonder de messages (spammer) des boîtes aux lettres. Les serveurs sont généralement configurés pour empêcher le relayage. Dans Postfix, cette option est configurée par défaut.

Le relayage pose plusieurs problèmes. Remplissage des boîtes aux lettres sans l’accord des destinataires, utilisation des ressources disques et CPU à l’insu des sociétés qui relaient les courriers...

Afin de combattre un peu le phénomène, une société qui relai les messages peut se voir "black listée", c’est à dire inscrite dans une liste noire référencée. Il existe plusieurs sites référençant ces listes noires. Certains de ces messages ne seront plus distribués. Voir pour cela : http://mail-abuse.org/rbl/, page principale de MAPS (Mail Abuse Prevention System LLC) RBLSM (Realtime Blackhole List).

Il est possible d’utiliser ces bases de données poru empêcher le relayage, ou refuser de délivrer les messages d’un site "black listé".

```
maps_rbl_domains = rbl.maps.vix.com
maps_rbl_reject_code = 554
reject_maps_rbl
```

Vous allez activer la fonction de relayage sur votre serveur et tester son comportement. Modifiez la ligne :

```
relay_domains = $mydestination
```

par

```
relay_domains = $mydestination, domaine1.dom, domaine2.dom...
```

où domaine1.dom, domaine2.dom...représentent les différents domaines de votre salle de TP. Relancer les services serveurs.

Vous pouvez maintenant à partir d’un client, utiliser le serveur smtp d’un autre domaine pour vous en servir comme "agent de relai" et envoyer des messages aux utilisateurs des autres domaines.

3.12. Autres techniques de filtrage et autres services de postfix

Le fichier de configuration main.cf, permet de filtrer sur les entêtes de messages (grep) sous sur le contenu (body). Ces outils permettent dans certains cas de limiter le spam.

Le serveur postfix.org tient à jour des produits complémentaires qui permettent de mettre en place des antivirus, des outils de filtrage de spam ou des outils de type web-mail.

Installation d’un serveur DDNS avec bind et DHCP

1. Résumé

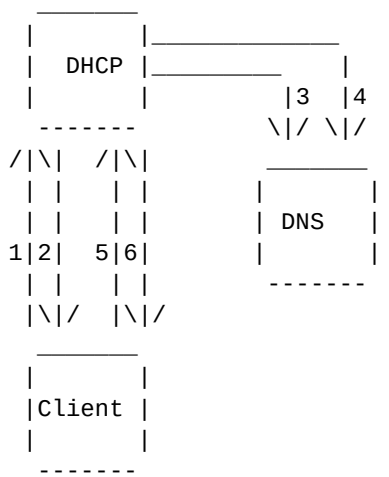
DHCP offre la possibilité de mettre à jour dynamiquement le système de résolution de nom.

Il s’agit, dans cette application, de faire cohabiter et faire fonctionner ensemble le service de résolution de nom bind et le service dhcp.

L’environnement a été testé sur une distribution debian, avec bind9 et dhcp3.

Vous devez savoir configurer un serveur DHCP, un serveur de nom, avoir compris le fonctionnement de rndc et des clés partagées, de dig.

Pour les amateurs d’ASCII-art, voici un schéma qui décrit les processus mis en oeuvre.




```
(1) DHCPDISCOVER from 00:08:c7:25:bf:5a (saturne) via eth0
(2) DHCPOFFER on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
(3) Added new forward map from saturne.freeduc-sup.org to 192.168.0.195
    Ajout de l’enregistrement de type A
(4) added reverse map from 195.0.168.192.in-addr.arpa to saturne.freeduc-sup.org
    Ajout de l’enregistrement de type PTR
(5) DHCPREQUEST for 192.168.0.195 (192.168.0.1) from 00:08:c7:25:bf:5a (saturne) via eth0
(6) DHCPACK on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
```

Les opérations 1, 2, 5, 6 ont déjà été vues lors de l'étude du service DHCP. On voit en étudiant le "log" ci-dessus, que l'inscription dans le DNS d'un client se fait avant l'acceptation du bail et l'inscription finale de ce client. (DHCPACK).

Dans cette application, vous installerez successivement le serveur de nom, le serveur dhcp, puis vous ferez les manipulations qui permettent l'intégration.

2. Éléments sur le service DDNS

Tout est décrit dans las pages de man de dhcpd.conf.

Deux façons de faire sont décrites (ad-hoc et interim) et une troisième est en cours d'élaboration. La méthode ad-hoc n'est semble t-il plus supportée par les paquets, du moins elle ne l'est pas avec le paquet dhcp3 de debian que j'utilise car considérée comme obsolète.

Le processus utilisé est défini par la variable, ddns-updates-style. Si la mise à jour n'est pas dynamique, la variable prend la valeur "none", nous, nous utiliserons "interim".

La méthode "ad-hoc" ne prend pas en charge le protocole "failover" des DHCP. C'est à dire qu'avec cette méthode vous ne pourrez pas avoir 2 serveurs DHCP assurant un système redondant et mettant à jour un même ensemble d'enregistrements DNS.

Le serveur détermine le nom du client en regardant d'abord dans les options de configuration des noms (ddns-hostname). Il est possible de générer dynamiquement un nom pour le client en concaténant des chaînes "dyn+N°+NomDeDomaine". S'il ne trouve rien, il regarde si le client lui a fait parvenir un nom d'hôte. Si aucun nom n'est obtenu, la mise à jour du DNS n'a pas lieu.

Pour déterminer le nom FQDN, le serveur concatène le nom de domaine au nom d'hôte du client.

Le nom du domaine lui, est défini uniquement sur le serveur DHCP.

Actuellement le processus ne prend pas en charge les clients ayant plusieurs interfaces réseau mais cela est prévu. Le serveur met à jour le DNS avec un enregistrement de type A et un enregistrement de type PTR pour la zone reverse. Nous verrons qu'un enregistrement de type TXT est également généré.

Quand un nouveau bail est alloué, le serveur crée un enregistrement de type "TXT" qui est une clé MD5 pour le client DHCP (DHCID).

La méthode "interim" est le standard. Le client peut demander au serveur DHCP de mettre à jour le serveur DNS en lui passant ses propres paramètres (nom FQDN). Dans ce cas le serveur est configuré pour honorer ou pas la demande du client. Ceci se fait avec le paramètre : "ignore client-updates" ou "allow client-updates".

Par exemple, si un client "jschmoe.radish.org" demande à être inscrit dans le domaine "exemple.org" et que le serveur DHCP est configuré pour, le serveur ajoutera un enregistrement PTR pour l'adresse IP mais pas

d’enregistrement A. Si l’option "ignore client-updates" est configuré, il y aura un enregistrement de type A pour "jschmoe.exemple.org".

3. Les aspects sur la sécurité

Le serveur DNS doit être configuré pour pouvoir être mis à jour par le serveur DHCP. La méthode la plus sûre utilise les signatures TSIG, basées sur une clé partagée comme pour le programme d’administration des serveurs de nom "rndc".

Vous devrez en créer une. Pour cela utiliser les éléments fournis dans la partie traitant de bind. Ces aspects y ont déjà été abordés.

Par exemple dans le fichier named.conf, le serveur DHCP disposant de la clé "DHCP_UPDATER", pourra mettre à jour la zone directe et la zone reverse pour lesquelles la déclaration "allow-update" existe.

Description du fichiers named.conf :

```
key DHCP_UPDATER {
 algorithm HMAC-MD5.SIG-ALG.REG.INT;
 secret pRP5FapFoJ95JEL06sv4PQ==;
       };

       zone "example.org" {
    type master;
    file "example.org.db";
    allow-update { key DHCP_UPDATER; };
       };

       zone "17.10.10.in-addr.arpa" {
    type master;
    file "10.10.17.db";
    allow-update { key DHCP_UPDATER; };
       };
```

Dans le fichier de configuration du serveur DHCP vous pourrez mettre :

```
 key DHCP_UPDATER {
 algorithm HMAC-MD5.SIG-ALG.REG.INT;
 secret pRP5FapFoJ95JEL06sv4PQ==;
       };

       zone EXAMPLE.ORG. {
 primary 127.0.0.1; # Adresse du serveur de noms primaire
 key DHCP_UPDATER;
       }

       zone 17.127.10.in-addr.arpa. {
 primary 127.0.0.1; # Adresse du serveur de noms primaire
 key DHCP_UPDATER;
       }
```

La clé DHCP_UPDATER déclarée pour une zone dans le fichier dhcpd.conf est utilisée pour modifier la zone si la clé correspond dans le fichier named.conf.

Les déclarations de zone doivent correspondre aux enregistrement SOA des fichiers de ressources des zones.

Normalement il n'est pas obligatoire d'indiquer l'adresse du serveur de nom primaire, mais cela peut ralentir le processus d'inscription des enregistrements, voire même ne pas fonctionner, si le serveur de nom n'a pas répondu assez vite.

4. Réalisation

Vous allez réaliser l'opération avec un client windows 2000 serveur et un client Linux. Le serveur Linux sera également serveur de nom.

Vous pouvez utiliser les exemples de fichiers fournis. Vosu aurez bien sûr à les adapter à votre configuration. Voici comment vont se dérouler les étapes :

1. Installation du serveur de nom et test
2. Installation du serveur DHCP et test
3. Intégration des deux services

Nous verrons à la fin comment générer des noms dynamiquement pour les clients.

5. Les fichiers de configuration

Dans les fichiers il y a des lignes qui sont en commentaires avec "###", elles seront décommentées pour la phase d'intégration des services

5.1. Le fichier named.conf

```
// Pour journaliser, les fichiers doivent créés
logging {
    channel update_debug {
 file "/var/log/log-update-debug.log";
 severity  debug 3;
 print-category yes;
 print-severity yes;
 print-time yes;
   };
   channel security_info    {
 file "/var/log/log-named-auth.info";
 severity  info;
 print-category yes;
 print-severity yes;
 print-time yes;
   };
```

```
    category update { update_debug; };
    category security { security_info; };
       };


// clé partagée entre bind, rndc et dhcp
include "/etc/bind/mykey";

options {
directory "/var/cache/bind";
query-source address * port 53;
auth-nxdomain yes;    # conform to RFC1035
forwarders { 127.0.0.1; 192.168.0.1;};
};

// Autorisations rndc sur la machine.
controls {
   inet  127.0.0.1   allow {any;} keys {mykey;};
   inet  192.168.0.0 allow {any;} keys {mykey;};
};


zone "." {
type hint;
file "/etc/bind/db.root";
};


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

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

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

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

zone "freeduc-sup.org" {
type master;
file "/etc/bind/freeduc-sup.org.hosts";
// Sert à la mise à jour par DHCP
// Sera décommenté lors de l'intégration des services
###        allow-update { key mykey; };
};
```

```
zone "0.168.192.in-addr.arpa" {
        type master;
        file "/etc/bind/freeduc-sup.org.hosts.rev";
// Sert à la mise à jour par DHCP
// Sera décommenté lors de l’intégration des services
###        allow-update { key mykey; };
        };
```

5.2. Le fichier de zone directe

```
$ORIGIN .
$TTL 86400 ; 1 day
freeduc-sup.org IN SOA master.freeduc-sup.org. root.freeduc-sup.org. (
2004050103 ; serial
10800      ; refresh (3 hours)
3600       ; retry (1 hour)
604800     ; expire (1 week)
38400      ; minimum (10 hours 40 minutes)
)
NS master.freeduc-sup.org.
MX 10 master.freeduc-sup.org.
$ORIGIN freeduc-sup.org.
master A 192.168.0.1
www CNAME master
```

5.3. Le fichier de zone in-addr.arpa

```
$ORIGIN .
$TTL 86400 ; 1 day
0.168.192.in-addr.arpa IN SOA master.freeduc-sup.org. root.freeduc-sup.org. (
2004050103 ; serial
10800      ; refresh (3 hours)
3600       ; retry (1 hour)
604800     ; expire (1 week)
38400      ; minimum (10 hours 40 minutes)
)
NS master.freeduc-sup.org.
$ORIGIN 0.168.192.in-addr.arpa.
1 PTR master.freeduc-sup.org.
```

5.4. Le fichier rndc.conf

```
include "/etc/bind/mykey";
options {
        default-server  localhost;
        default-key     "mykey";
};

server localhost {
        key     "mykey";
};
```

5.5. Le fichier de clé partagée

Ici il est nommé "mykey".

```
key "mykey" {
        algorithm       hmac-md5;
        secret "X/ErbPNOiXuC8MIgTX6iRcaq/1OFCEDIlxrmnfPgdqYIOY3U6lsgDMq15jnxXEXmdGvv1g/ayYtAA73bU
};
```

5.6. Le fichier dhcpd.conf

```
ddns-update-style none;
### ddns-update-style interim;
###          deny client-updates;
###          ddns-updates on;
###          ddns-domainname "freeduc-sup.org";
###          ddns-rev-domainname "in-addr.arpa";
authoritative;

subnet 192.168.0.0 netmask 255.255.255.0 {
option broadcast-address  192.168.0.255;
option routers 192.168.0.2;
option domain-name "freeduc-sup.org";
option domain-name-servers 192.168.0.1;
option broadcast-address  192.168.0.255;
option routers 192.168.0.2;
range 192.168.0.100 192.168.0.195;
default-lease-time 600;
max-lease-time 7200;

# Instructions pour la mise à jour des zones
### include "/etc/bind/mykey";

### zone freeduc-sup.org. {
###         primary 192.168.0.1;
###         key mykey;
###         }

### zone 0.168.192.in-addr.arpa. {
###         primary 192.168.0.1;
###         key mykey;
###         }

}
```

6. Procédure de tests des services

Vous allez pouvoir tester. À partir de maintenant vous devrez consulter les fichiers de logs si vous rencontrez des problèmes de fonctionnement, les tables de processus... bref tout ce qui pourra vous permettre de déterminer la ou les sources possibles des dysfonctionnements si vous en constatez.

Lancez le service bind et tester son fonctionnement avec rndc.

```
root@master:/home/knoppix# rndc status
number of zones: 8
debug level: 0
xfers running: 0
xfers deferred: 0
soa queries in progress: 0
query logging is OFF
server is up and running
root@master:/home/knoppix#
```

Ça permet de vérifier que la clé est bien reconnue.

Vérifiez le fonctionnement du serveur à l'aide de la commande dig.

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 freeduc-sup.org axfr
; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
;; Query time: 36 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:15:38 2003
;; XFR size: 8 records
```

Vérifier de la même façon le fonctionnement de la zone reverse.

Vérifiez la strucure du fichier dhcp.

```
root@master:/home/knoppix# dhcpd3 -t
Internet Software Consortium DHCP Server V3.0.1rc9
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
```

Ça permet de vérifier qu'il n'y a pas d'erreur de syntaxe dans le fichier.

Testez le fonctionnement traditionnel de votre serveur DHCP à partir d'un client Linux et windows. Faites des renouvellement de baux.

7. Intégration des services

Par défaut les client Linux ne transmettent pas leur nom d'hôte comme c'est le cas pour les clients windows. Modifiez sur le client Linux le fichier /etc/dhclient.conf de la façon suivante, nous verrons plus loin comment générer un nom dynamiquement :

```
[root@bestof mlx]# more /etc/dhclient.conf
           send host-name "bestof";
```

Décommentez dans les fichiers named.conf et dhcpd.conf les lignes commentées par "###".

Supprimez dans le dhcpd.conf la ligne :

```
ddns-update-style none;
```

Relancez le service DNS et testez sont bon fonctionnement

Vérifiez le fichier dhcpd.conf avec la commande dhcpd3 -t

Lancez dhcp en mode "foreground" dhcpd3 -d, voici ce que vous devriez obtenir :

```
root@master:/etc/dhcp3# dhcpd3 -d
Internet Software Consortium DHCP Server V3.0.1rc9
Copyright 1995-2001 Internet Software Consortium.
All rights reserved.
For info, please visit http://www.isc.org/products/DHCP
Wrote 1 leases to leases file.
Listening on LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   LPF/eth0/00:d0:59:82:2b:86/192.168.0.0/24
Sending on   Socket/fallback/fallback-net
```

Demandez un bail à partir du client windows (ici windows 2000 Server) , voici ce qui devrait se passer :

```
DHCPDISCOVER from 00:08:c7:25:bf:5a (saturne) via eth0
DHCPOFFER on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
Added new forward map from saturne.freeduc-sup.org to 192.168.0.195
added reverse map from 195.0.168.192.in-addr.arpa to saturne.freeduc-sup.org
DHCPREQUEST for 192.168.0.195 (192.168.0.1) from 00:08:c7:25:bf:5a (saturne) via eth0
DHCPACK on 192.168.0.195 to 00:08:c7:25:bf:5a (saturne) via eth0
```

Demandez un bail à partir du client Linux, voici ce qui devrait se passer :

```
DHCPDISCOVER from 00:08:c7:25:ca:7c via eth0
DHCPOFFER on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
Added new forward map from bestof.freeduc-sup.org to 192.168.0.194
added reverse map from 194.0.168.192.in-addr.arpa to bestof.freeduc-sup.org
DHCPREQUEST for 192.168.0.194 (192.168.0.1) from 00:08:c7:25:ca:7c (bestof) via eth0
DHCPACK on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
```

Voici le contenu du fichier de journalisation de bind :

```
Log de Bind log-update-debug.log
root@master:/var/log# more log-update-debug.log
May 06 07:49:50.457 update: info: client 192.168.0.1#32846: updating zone 'freeduc-sup.org/IN': ad
May 06 07:49:50.458 update: info: client 192.168.0.1#32846: updating zone 'freeduc-sup.org/IN': ad
May 06 07:49:50.512 update: info: client 192.168.0.1#32846: updating zone '0.168.192.in-addr.arpa/
t
May 06 07:49:50.512 update: info: client 192.168.0.1#32846: updating zone '0.168.192.in-addr.arpa/
May 06 07:50:47.011 update: info: client 192.168.0.1#32846: updating zone 'freeduc-sup.org/IN': ad
May 06 07:50:47.011 update: info: client 192.168.0.1#32846: updating zone 'freeduc-sup.org/IN': ad
May 06 07:50:47.017 update: info: client 192.168.0.1#32846: updating zone '0.168.192.in-addr.arpa/
t
May 06 07:50:47.017 update: info: client 192.168.0.1#32846: updating zone '0.168.192.in-addr.arpa/
root@master:/var/log#
```

Voici le contenu du fichier de déclaration de zone avec les nouveaux enregistrements

```
root@master:/var/log# dig @127.0.0.1 freeduc-sup.org axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 2004
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
bestof.freeduc-sup.org. 300     IN      TXT     "00e31b2921cd30bfad552ca434b61bda02"
bestof.freeduc-sup.org. 300     IN      A       192.168.0.194
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
saturne.freeduc-sup.org. 300    IN      TXT     "310e43cfc20efbe1c96798d48672bc76aa"
saturne.freeduc-sup.org. 300    IN      A       192.168.0.195
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 2004
;; Query time: 381 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 07:56:43 2003
;; XFR size: 12 records
```

8. Générer un nom dynamiquement pour les clients DHCP

Cela est possible en modifiant le fichier de configuration de DHCP. Vous pourrez retrouver tous les éléments dans la page de manuel.

Par exemple rajoutez dans le fichier la ligne ci-dessous pour adapter le nom à partir de l'adresse MAC du client :

```
#ddns-hostname = binary-to-ascii (16, 8, "-",substring (hardware, 1, 12));
```

Ou celle-ci pour localiser le client :

```
ddns-hostname = concat ("dhcp-a-limoges","-",binary-to-ascii(10,8,"-",leased-address));
```

Avec cette dernière, voici les enregistrements ajoutés :

```
Added new forward map from dhcp-a-limoges-192-168-0-194.freeduc-sup.org to 192.168.0.194
added reverse map from 194.0.168.192.in-addr.arpa to dhcp-a-limoges-192-168-0-194.freeduc-sup.org
DHCPREQUEST for 192.168.0.194 from 00:08:c7:25:ca:7c via eth0
DHCPACK on 192.168.0.194 to 00:08:c7:25:ca:7c (bestof) via eth0
```

Le fichier des incriptions :

```
root@master:/home/knoppix# more /var/lib/dhcp3/dhcpd.leases
lease 192.168.0.194 {
  starts 2 2003/05/06 17:38:38;
  ends 2 2003/05/06 17:48:38;
  binding state active;
  next binding state free;
  hardware ethernet 00:08:c7:25:ca:7c;
  set ddns-rev-name = "194.0.168.192.in-addr.arpa";
  set ddns-txt = "00e31b2921cd30bfad552ca434b61bda02";
  set ddns-fwd-name = "dhcp-192-168-0-194.freeduc-sup.org";
  client-hostname "bestof";
}
```

Les transferts de zones directes et inverses :

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 freeduc-sup.org axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 freeduc-sup.org axfr
;; global options:  printcmd
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
freeduc-sup.org.        86400   IN      NS      master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      MX      10 master.freeduc-sup.org.
0-8-c7-25-ca-7c.freeduc-sup.org. 300 IN TXT     "00e31b2921cd30bfad552ca434b61bda02"
0-8-c7-25-ca-7c.freeduc-sup.org. 300 IN A       192.168.0.194
argo.freeduc-sup.org.   86400   IN      A       192.168.0.253
dhcp-192-168-0-194.freeduc-sup.org. 300 IN TXT  "00e31b2921cd30bfad552ca434b61bda02"
dhcp-192-168-0-194.freeduc-sup.org. 300 IN A    192.168.0.194
dhcp-a-limoges-192-168-0-194.freeduc-sup.org. 300 IN TXT "00e31b2921cd30bfad552ca434b61bda02"
dhcp-a-limoges-192-168-0-194.freeduc-sup.org. 300 IN A 192.168.0.194
master.freeduc-sup.org. 86400   IN      A       192.168.0.1
www.freeduc-sup.org.    86400   IN      CNAME   master.freeduc-sup.org.
freeduc-sup.org.        86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:39:08 2003
;; XFR size: 14 records
```

La zone reverse :

```
root@master:/home/knoppix/tmp# dig @127.0.0.1 0.168.92.in-addr.arpa axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 0.168.92.in-addr.arpa axfr
;; global options:  printcmd
; Transfer failed.
root@master:/home/knoppix/tmp# dig @127.0.0.1 0.168.192.in-addr.arpa axfr

; <<>> DiG 9.2.2 <<>> @127.0.0.1 0.168.192.in-addr.arpa axfr
```

```
;; global options:  printcmd
0.168.192.in-addr.arpa. 86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
0.168.192.in-addr.arpa. 86400   IN      NS      master.freeduc-sup.org.
1.0.168.192.in-addr.arpa. 86400 IN      PTR     master.freeduc-sup.org.
194.0.168.192.in-addr.arpa. 300 IN      PTR     dhcp-192-168-0-194.freeduc-sup.org.
3.0.168.192.in-addr.arpa. 86400 IN      PTR     argo.freeduc-sup.org.
0.168.192.in-addr.arpa. 86400   IN      SOA     master.freeduc-sup.org. root.freeduc-sup.org. 200
;; Query time: 3 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Tue May  6 19:40:08 2003
;; XFR size: 7 records
```

Installation d’un service Web-mail

Le service Web-mail.

1. Présentation

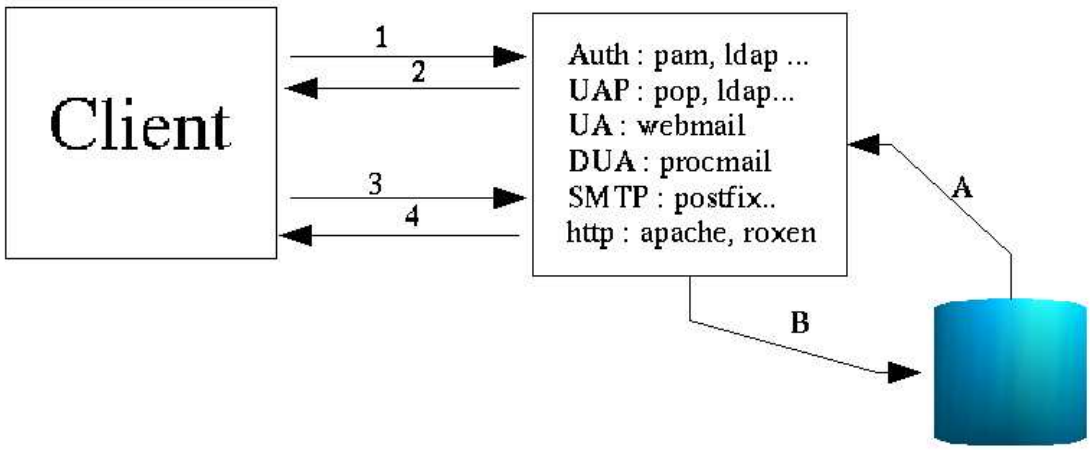
Il est préférable d’avoir réalisé les ateliers sur les serveurs HTTP, SMTP et DNS avant de commencer celui-ci.

Le service Web-mail permet l’utilisation d’un service de messagerie à partir d’un client Web comme mozilla. Cette interface est intéressante, car contrairement à un client pop3 standard qui serait configuré pour rapatrier les courriers sur la machine locale, ceux-ci, vont rester sur le serveur. Vous pouvez les consulter à partir de n’importe quel poste pourvu qu’il dispose d’un navigateur. Vous aurez ensuite tout le loisir de les récupérer avec votre client de messagerie préféré si vous en utilisez un.

Il existe un très grand nombre de serveur Web-mail, et écris dans des langages très différents. Une étude a été réalisée par le CRU http://www.cru.fr/http-mail/, mais parmi les principaux on peut citer IMP écris en PHP et qui s’appuie sur la librairie horde. Il existe aussi OpenWebmail qui lui est écris en Perl. Ces deux produits existent en paquets Debian, on utilisera pour le TP OpenWebmail, mais ces deux outils comportent chacun de nombreuses qualités, le choix devra se faire en fonction du degré d’intégration que vous souhaiterez obtenir avec vos autres applications.

2. Architecture générale du service

Figure 20. Architecture globale d’un service Web-mail



1. En 1 et 2, le client passe par une phase préalable d’authentification, il faut donc un service correspondant sur le serveur. Cela peut être pris directement en charge par le service Web-mail ou par un service extérieur (pam, ldap par exemple). (Nous utiliserons l’authentification pam)
2. En 3 et 4, le dialogue s’effectue en un navigateur et un serveur HTTP. Le dialogue peut s’effectuer dans un canal SSL ou TLS. Le suivi de session peut être réalisé à l’aide de cookie par exemple. (Nous utiliserons Apache comme serveur HTTP).
3. En A et B, le service Web-mail utilise une base de données pour les Boîtes aux lettres des utilisateurs, pour les dossiers (inbox, outbox, trash...) et la conservation des courriers. Certains Web-mail s’appuient sur des bases de type MySQL, PostgreSQL... ou simplement sur une arborescence de répertoires dans le HOME_DIRECTORY de l’utilisateur. (Nous n’utiliserons pas de SGBD/R).
4. Sur le serveur, il faut activer un protocole de traitement du courrier (pop3, pop3s, imap, imaps...). Nous utiliserons imap et pop3.
5. Vous aurez également besoin d’un service SMTP pour le traitement des courriers sortants (nous utiliserons postfix) et d’un service de livraison (DUA) (nous utiliserons procmail) pour délivrer les courriers entrants.

3. Installation et confi guration OpenWebmail

Vous allez installer "OpenWebmail" mais auparavant il est nécessaire de s’assurer du bon fonctionnement de certains services. (smtp, mail, procmail, apache...)

3.1. Préparation de la machine

Suivez la procédure ci-dessous pour préparer la machine.

3.1.1. Configuration générale

On considère la configuration suivante, vous adapterez les noms, adresses ip et autres paramètres à votre configuration. Il n'y a pas de DNS.

```
Nom d'hôte : freeduc-sup ($NAME)
Nom FQDN de la machine : freeduc-sup.foo.org ($FQDN)
Adresse de réseau : 192.168.0.0
Adresse de la machine : 192.168.0.2
Adresse de la passerelle par défaut : 192.168.0.254
```

3.1.2. Test de la résolution de nom

Vérifier que la résolution de nom fonctionne parfaitement. Les commandes : ping $NAME et ping $FQDN doivent répondre correctement.

```
# Exemple de fichier /etc/hosts
127.0.0.1       freeduc-sup freeduc-sup.foo.org localhost localhost.localdomain
192.168.0.2    freeduc-sup  freeduc-sup.foo.or
```

3.1.3. Le service Apache

Activez le service apache. Il ne doit pas y avoir de message d'erreur au lancement, notamment sur la résolution de nom. Vérifiez également le bon fonctionnement avec une requête sur : http://localhost

3.1.4. Le service SMTP (Postfix)

Pour configurer postfix, utilisez la commande "dpkg-reconfigure postfix", vous prendrez site internet. Les valeurs par défaut doivent normalement fonctionner.

Ouvrez le fichier /etc/postfix/main.cf, vérifiez qu'il correspond à celui-ci, au besoin modifiez le :

```
# Fichier de configuration de Postfix
# Adaptez vos noms d'hôtes et vos noms de machines

# see /usr/share/postfix/main.cf.dist for a commented, fuller
# version of this file.

# Do not change these directory settings - they are critical to Postfix
# operation.
command_directory = /usr/sbin
daemon_directory = /usr/lib/postfix
```

```
program_directory = /usr/lib/postfix

smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
setgid_group = postdrop
biff = no

myhostname = freeduc-sup.foo.org
mydomain = foo.org
myorigin = $myhostname
inet_interfaces = all
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
mydestination = $myhostname, localhost.$mydomain
relayhost =
mailbox_command = procmail -a "$EXTENSION"
mailbox_size_limit = 0
recipient_delimiter = +
```

Une fois cela réalisé, activez ou relancez le service.

```
/etc/init.d/postfix start
/etc/init.d/postfix reload
```

3.1.5. Activation des services imap

Cela s'effectue dans le fichier inet.conf. Il faudra adapter si vous utilisez xinetd. Cela dépend de la distribution de GNU/Linux que vous utilisez. Vous pouvez également utiliser la commande "dpkg-reconfigure uw-imapd". Dans ce cas, prenez imap2 (qui correspond à imap4 (? ;-))) et imaps.

Extrait d'un exemple de configuration de inetd.conf :

```
#:MAIL: Mail, news and uucp services.
imap2   stream  tcp     nowait  root    /usr/sbin/tcpd /usr/sbin/imapd
```

Adaptez votre fichier de configuration, puis relancer inetd avec la commande :

```
/etc/init.d/inetd restart
```

Vérifiez que les ports sont bien ouverts avec la commande "netstat". Vous devez avoir les ports 25 (pop3)si vous l'avez activé et 143 (imap) ouverts.

```
netstat -atup | grep LISTEN
tcp     0      0 *:netbios-ssn       *:*      LISTEN      269/smbd
tcp     0      0 *:imap2             *:*      LISTEN      263/inetd
tcp     0      0 *:sunrpc            *:*      LISTEN      151/portmap
tcp     0      0 *:ssh               *:*      LISTEN      278/sshd
```

```
tcp     0      0 *:ipp                   *:*       LISTEN      290/cupsd
tcp     0      0 *:smtp                  *:*       LISTEN      899/master
```

Nous voyons imap2, ligne 2, pris en charge par inetd.

```
root@freeduc-sup:/home/mlx# netstat -natup | grep LISTEN
tcp   0      0 0.0.0.0:139   0.0.0.0:*       LISTEN      269/smbd
tcp   0      0 0.0.0.0:143   0.0.0.0:*       LISTEN      263/inetd
tcp   0      0 0.0.0.0:111   0.0.0.0:*       LISTEN      151/portmap
tcp   0      0 0.0.0.0:22    0.0.0.0:*       LISTEN      278/sshd
tcp   0      0 0.0.0.0:631   0.0.0.0:*       LISTEN      290/cupsd
tcp   0      0 0.0.0.0:25    0.0.0.0:*       LISTEN      899/master
```

Ici l’option "-n" de netstat nous indique les numéros de ports utilisés.

Remarque : si vous souhaitez utiliser un client pop, vous devrez activer également le protocole pop.

3.1.6. Test des services

À ce stade, il nous est possible de tester complètement le service de messagerie. Vous pouvez indifféremment utiliser un client comme kmail ou Mozilla. Le plus simple est d’utiliser le client "mail".

Suivez la procédure ci-dessous :

1. Créez deux comptes utilisateurs alpha et beta qui serviront pour les tests avec la commande "adduser".
2. Testez avec la commande "mail" que l’envoie de courrier se déroule correctement. Les commandes :

   ```
   mail alpha
   mail alpha@freeeduc-sup
   mail alpha@freeduc-sup.foo.org
   ```

   doivent fonctionner correctement.

3.2. Installation d’OpenWebmail

Si vous utilisez la freeduc-sup, OpenWebmail n’est pas installé. Utilisez la commande :

```
apt-get install openwebmail
```

La commande installera également 3 paquets supplémentaires qui correspondent aux dépendances puis lancera la procédure de configuration.

3.3. Confi guration de l’application OpenWebmail

Vous pouvez à tout moment reconfigurer l’application avec "dpkg-reconfigure openwebmail". Prenez comme option :

```
authentification -> auth_pam.pl
langage -> fr
```

3.4. Test de l’environnement

Pour tester l’environnement vous avez deux liens :

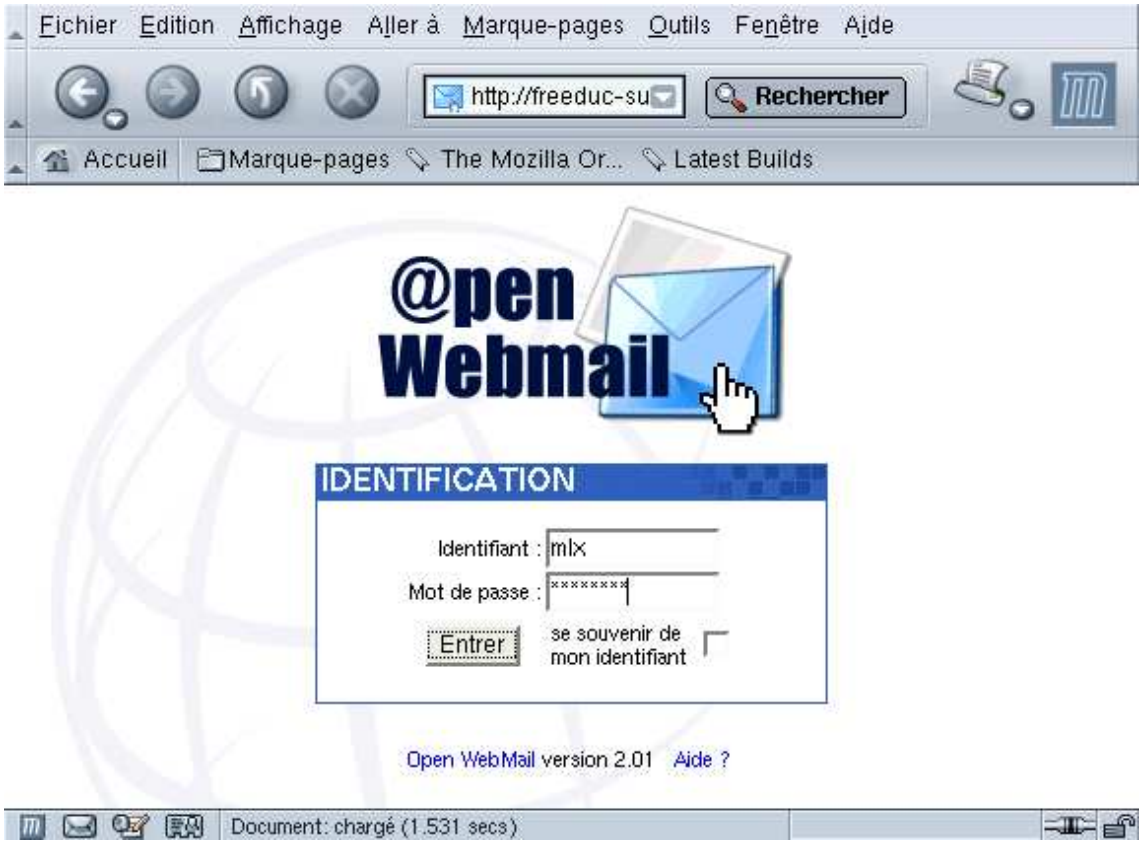
`http://localhost/openwebmail`

qui vous place sur un espace documentaire

`http://localhost/cgi-bin/openwebmail/openwebmail.pl`

qui lance l’application proprement dite et vous amène sur la première fenêtre de login

Figure 21. Ouverture de session sur un Web-mail

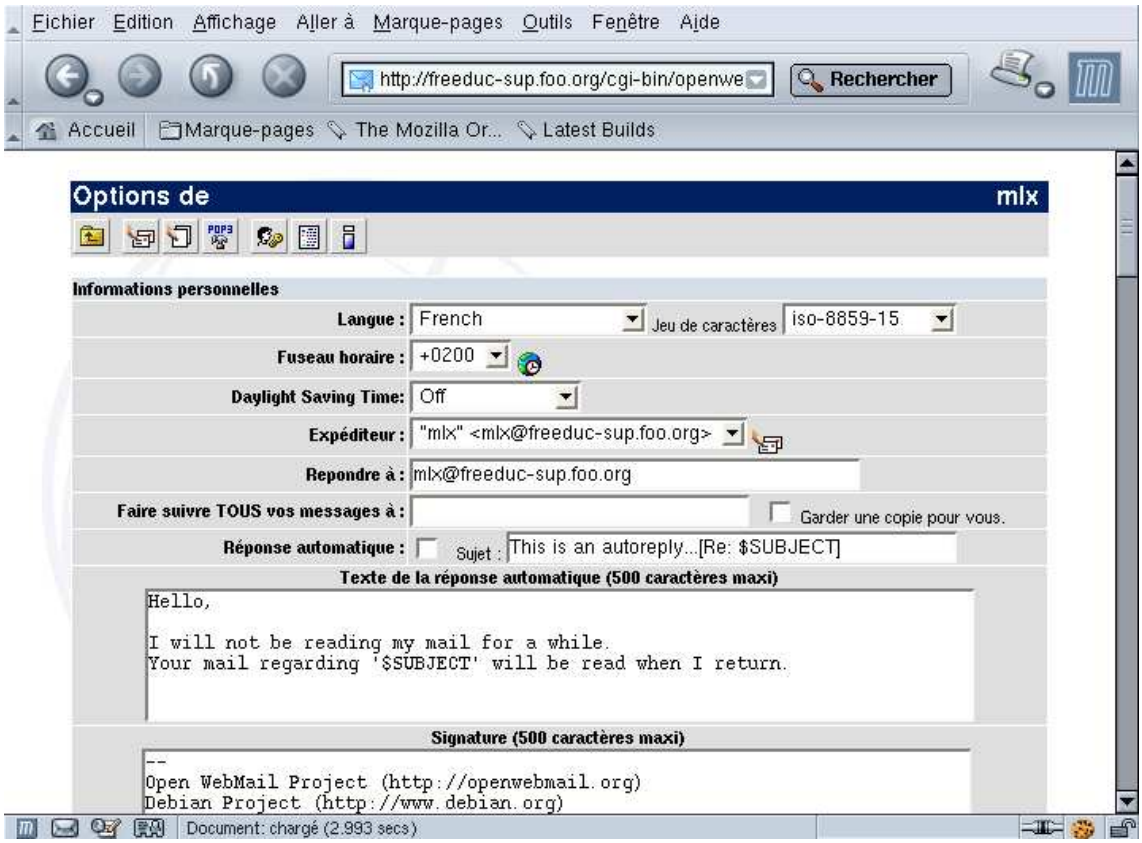


Il vous sera possible de créer un serveur Web virtuel pour avoir par exemple : "openwebmail.freeduc-sup.org".

3.5. Confi guration de l'environnement utilisateur

À la première session, la personne reçoit une invite lui permettant de configurer son environnement et ses paramètres particuliers, comme son adresse de réponse, modifier son mot de passe... Ces paramètres sont modifiables à tout moment.

Figure 22. Configuration de l'environnement utilisateur



3.6. Test et environnement OpenWebmail

À partir de ce moment, l'environnement complet est disponible. L'utilisateur dispose également d'un calendrier et d'une documentation en ligne.

Figure 23. Voir ses messages

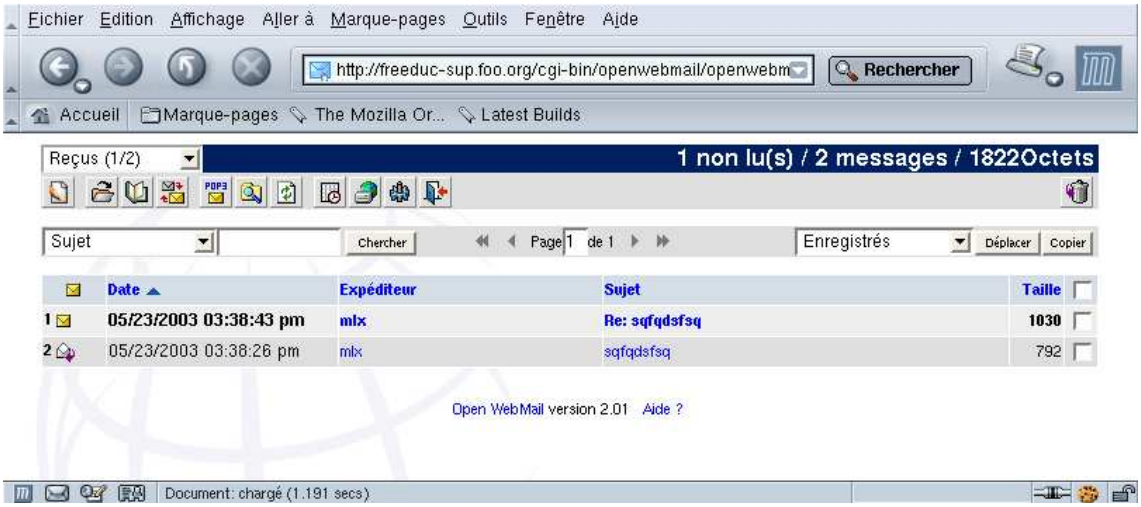


Figure 24. Le calendrier

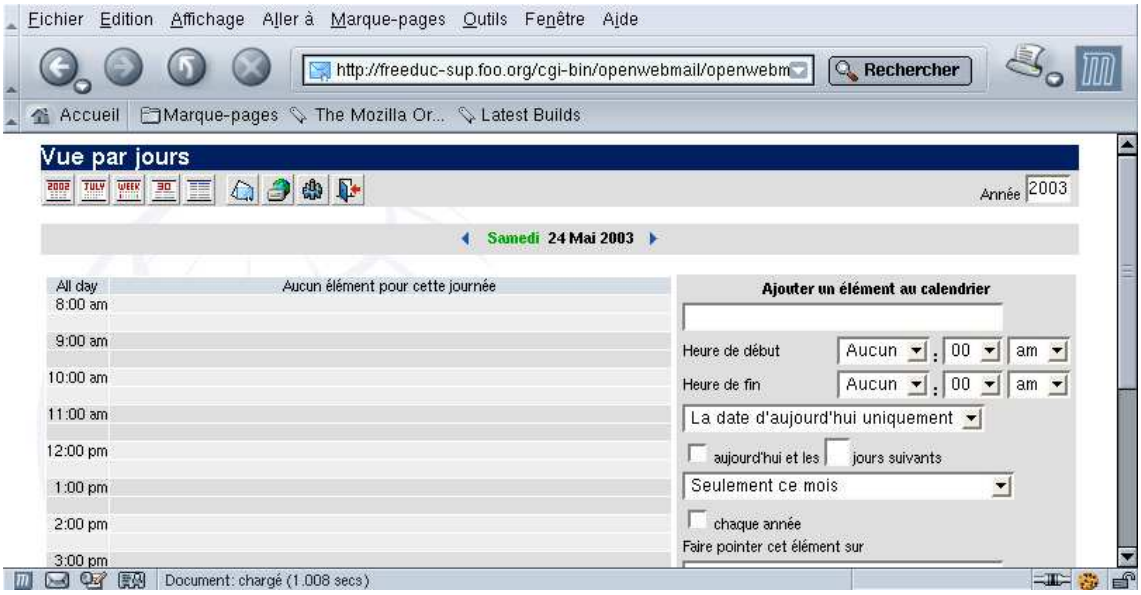


Figure 25. L'aide en ligne

4. Application

1. Configurez et vérifiez le bon fonctionnement des services de résolution de nom, apache, postfix.
2. Installez et configurez OpenWebmail.
3. Testez le fonctionnement OpenWebmail.

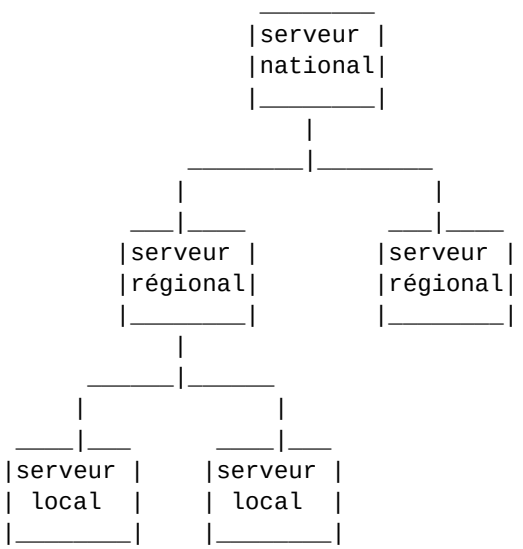
Installation d’un service mandataire (Proxy SQUID)

Le service proxy. Installation et configuration de SQUID.

1. Présentation

Squid est un service serveur proxy-cache sous linux. Les objets consultés par les clients sur internet, sont stockés en cache disque par le serveur. À partir du deuxième accès, la lecture se fera en cache, au lieu d’être réalisée sur le serveur d’origine. De ce fait il permet "d’accélérer" vos connexions à l’internet en plaçant en cache les documents les plus consultés. On peut aussi utiliser la technique du service serveur mandataire pour effectuer des contrôles d’accès aux sites.

Les services proxy peuvent être organisés de façon hiérarchique :



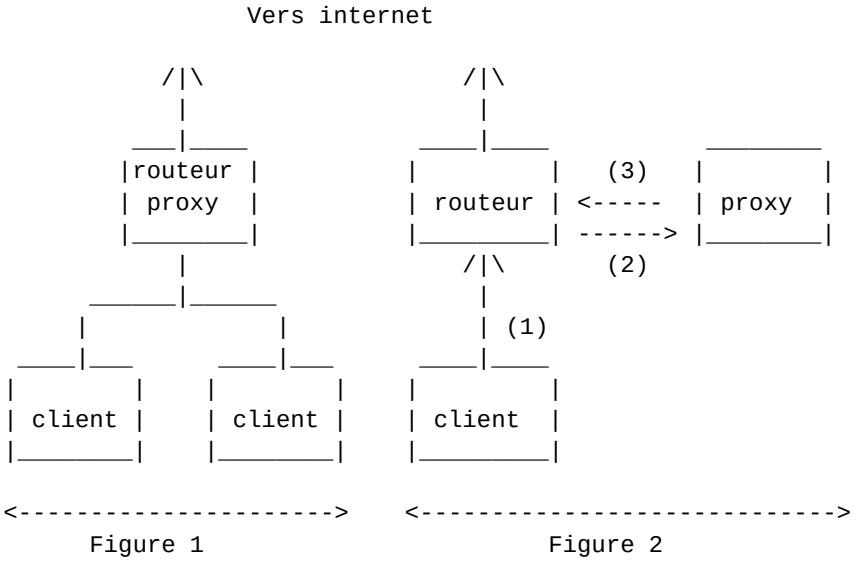
Ls serveurs peuvent être paramétrés pour les autorisations d’accès et les synchronisations.

Les postes clients sont souvent configurés pour utiliser un serveur proxy. Le client s’adresse au serveur proxy, et c’est ce dernier qui traite la requête sur internet. Un fois la réponse reçue, le serveur met en cache la réponse et la

retourne au client interne. Le service proxy est fréquemment configuré sur un routeur qui remplit aussi le service de translation d’adresse ou translation de port, mais toutes ces fonctions sont bien différentes.

Dans certains cas, on peut ne pas souhaiter que la configuration soit réalisée au niveau du client. On souhaite que celle-ci soit faite au niveau du serveur. Cela peut arriver par exemple si vous avez plusieurs centaines de postes à configurer ou bien si vous ne souhaitez pas que les utilisateurs puissent modifier ou avoir accès à cette partie de la configuration. On parlera de "service proxy transparent". Le service serveur proxy peut être sur le routeur d’accès à l’internet ou sur une autre machine.

```
Service proxy tranparent :
La configuration des navigateurs, sur les postes clients, n’est pas concernée.
```



Sur la Figure 1, le service proxy est installé sur le routeur.

Sur la figure 2, les requêtes du client (1), sont redirigées vers le proxy par le routeur (2), qui retourne au client la réponse ou redirige vers le routeur (3) pour un envoie sur l’extérieur.

1.1. Installer Squid

Sur debian "apt-get install squid".

Squid comporte de très nombreux paramètres. L’optimisation n’en est pas toujours simple. Nous allons voir uniquement quelques options permettant un fonctionnement du service. Il sera nécessaire, pour un site en production, de se référer à la documentation officielle.

Pour démarrer une configuration simple, il est possible d’utiliser le fichier de configuration /etc/squid.conf, dont chaque paramètre est documenté.

1.2. Confi guration de squid

Toute la configuration de Squid se trouve dans le fichier "squid.conf". La plupart des options par défaut du fichier ne sont pas à changer (vous pouvez alors laisser le # pour conserver les options en commentaire.)

http_port: le port que vous souhaitez utiliser. Le plus fréquent est 8080. Il faut donc changer cette valeur car par défaut Squid utilise 3128.

icp_port: conserver le port 3130. Ceci vous permet de communiquer avec des proxy-cache parents ou voisins.

cache_mem : correspond au cache mémoire, la valeur dépend de votre système. Par défaut squid utilise 8 Mo. Cette taille doit être la plus grande possible afin d’améliorer les performances (Considérez 1/3 de la mémoire que vous réservez à Squid). Il faut avec cache_mem régler cache_mem_low et cache_mem_high qui sont les valeurs limites de remplissage du cache mémoire. Par défaut les valeurs sont 75 % et 90 %. Lorsque la valeur de 90 % est atteinte le cache mémoire se vide jusqu’à 75 %. Les valeurs par défaut sont correctes dans la plupart des cas.

cache_swap : correspond à la taille de votre cache disque. Si la taille du disque le permet, et en fonction de la taille de votre établissement (nombre de client qui utilise le cache), mais aussi de la durée de rafraîchissement de votre cache et du débit de votre ligne, vous devez mettre la valeur qui vous semble correspondre à votre situation.

acl QUERY urlpath_regex cgi-bin \? \.cgi \.pl \.php3 \.asp : Type de page à ne pas garder dans le cache afin de pas avoir les données d’un formulaire par exemple.

maximum_object_size : taille maximale de l’objet qui sera sauvegardé sur le disque. On peut garder la valeur par défaut.

cache_dir : Vous indiquez ici le volume de votre cache. Si vous avez plusieurs disques utilisez plusieurs fois cette ligne. Si squid ne fonctionne pas bien, où s’arrête parfois sans raison apparente, vérifiez que vous avez un cache assez important ou bien configuré.

```
cache_dir ufs /cache1 100 16 256         (cache de 100 Mb)
cache_dir ufs /cache2 200 16 256         (cache de 200 Mb)
```

Les valeurs 16 et 256, indiquent le nombre de sous-répertoires créés respectivement dans le premier niveau et suivants pour le stockage des données du cache.

cache_access_log ; cache_log ; cache_store_log : Indique l’endroit ou se trouve les logs (fichiers de journalisation. Si vous ne souhaitez pas avoir de log (par exemple des objets cache_store_log) indiquer cache_store_log none.

debug_options ALL,1 : niveau de debug. Indiquer 9 pour avoir toutes les traces à la place de 1. Attention cela donne de gros fichiers.

dns_children : Par défaut le nombre de processus simultanés dns est de 5. Il peut être nécessaire d’augmenter ce nombre afin que Squid ne se trouve pas bloqué. Attention de ne pas trop l’augmenter cela pouvant poser des problèmes de performance à votre machine (indiquer 10 ou 15).

request_size : Taille maximale des requêtes. Conserver le défaut, concerne les requêtes de type GET, POST...

refresh_pattern : Permet de configurer la durée de mise à jour du cache. Utiliser "-i" pour ne pas tenir compte des minuscules ou des majuscules. (voir le fichier squid.conf). Les valeurs Min et Max sont indiquées en minutes. Exemple :

```
# refresh_pattern ^ftp:         1440    20%     10080
```

visible_hostname : indiquer ici le nom de votre serveur proxy.

logfile_rotate : Pour faire tourner vos logs et garder un nombre de copies. par défaut 10. attention si votre cache est trés utilisé il peut générer un grand volume de logs, pensez donc à réduire ce nombre.

error_directory : Pour avoir les messages d’erreurs en français (indiquer le répertoire ou ils se trouvent). Exemple :

```
#error_directory /etc/squid/errors
#Créer un lien vers le répertoire où sont logés les messages en Français.
```

1.3. Initialisation de Squid

Cela n’est réalisé que la première fois afin de générer le cache.

```
squid -z
```

1.4. Les options de démarrage de squid

On peut aussi démarrer squid en lui passant des commandes sur la ligne de commande. Différents paramètres peuvent être passés sur la ligne de commande. Les options passées de cette façon remplacent les paramètres du fichier de configuration de Squid : "squid.conf".

```
-h : Pour obtenir les options possibles
-a : Pour indiquer un port particulier
-f : pour utiliser un autre fichier de conf  au lieu de squid.conf
-u : spécifie un port pour les requêtes ICP. (3110 par défaut)
-v : pour indiquer la version de Squid
-z : Pour initialiser le disque cache.
-k : Pour envoyer des instructions à Squid pendant son fonctionnement.
     Il faut faire suivre -k d’une instruction
     (rotate|reconfigure|shutdown|interrupt|kill|debug|check).
-D : pour démarrer squid lorsque vous n’êtes pas connecté en
     permanence à internet (évite de vérifier si le serveur DNS répond).
```

1.5. Contrôler les accès

Pour contrôler tout ce qui passe par votre serveur proxy, vous pouvez utiliser ce que l’on appelle les ACL (Access Control List). Les ACL sont des règles que le serveur applique. Cela permet par exemple d’autoriser ou d’interdire certaines transactions.

On peut autoriser ou interdire en fonction du domaine, du protocole, de l’adresse IP, du numéro de port, d’un mot, on peut aussi limiter sur des plages horaires.

La syntaxe d’une ACL est la suivante :

```
acl             aclname          acltype             string[string2]
http_access     allow|deny       [!]aclname
```

acltype peut prendre comme valeur :

```
src (pour la source) : indication de l’adresse IP du client sous la
    forme adresse/masque. On peut aussi donner une plage d’adresse
    sous la forme adresse_IP_debut-adresse_IP_fin
dst (pour la destination) : idem que pour src, mais on vise
    l’adresse IP de l’ordinateur cible.
srcdomain : Le domaine du client
dstdomain : Le domaine de destination.
url_regex : Une chaîne contenu dans l’URL
   (on peut utiliser les jokers ou un fichier).
urlpath_regex : Une chaîne comparée avec le chemin de l’URL
   (on peut utiliser les jokers).
proto : Pour le protocole.
```

Exemple 1 : Interdire l’accés à un domaine : Supposons que nous souhaitions interdire l’accés à un domaine (par exemple le domaine pas_beau.fr). On a donc

```
acl           veuxpas         dstdomain         pas_beau.fr
http_access   deny            veuxpas
http_access   allow           all     # On accepte tout
```

La dernière ligne ne doit exister qu’une fois dans le fichier squid.conf.

Exemple 2 : Interdire l’accés aux pages contenant le mot jeu.

```
acl           jeu            url_regex           jeu
http_access   deny            jeu
http_access   allow           all
```

Attention url_regex est sensible aux majuscules/minuscules. Pour interdire JEU il faut aussi ajouter JEU dans votre ACL. Il n'est pas besoin de réécrire toute l'ACL. On peut ajouter JEU derrière jeu en laissant un blanc comme séparation (cela correspondant à l'opérateur logique OU).

On peut placer un nom de fichier à la place d'une série de mots ou d'adresses, pour cela donner le nom de fichier entre guillemets. Chaque ligne de ce fichier doit contenir une entrée.

Exemple 3 : Utilisation d'un fichier

```
# URL interdites
acl          url_interdites  url_regex "/etc/squid/denied_url"
http_access  deny            url_interdites
```

Des produits associés à Squid (redirecteurs) permettent un contrôle plus simple. SquidGuard, par exemple, permet d'interdire des milliers de sites. Le site d'information est référencé plus loin dans la rubrique "liens". Pensez, si vous utilisez SquidGuard, à configurer la ligne suivante dans le fichier squid.conf :

```
redirect_program /usr/local/squid/bin/SquidGuard
```

Exemple 4 : pour contrôler qui a le droit d'utiliser votre cache, créez une ACL du type :

```
acl             si_OK        src    192.168.0.0/255.255.0.0
http_access     allow        localhost
http_access     allow        site_OK
http_access     deny         all
```

1.6. Contrôler les accés par authentifi cation

Parmi les demandes qui reviennent le plus souvent, la question de l'utilisation de Squid pour contrôler qui a le droit d'aller sur internet, est l'une des plus fréquente.

On peut imaginer deux solutions :

La première consiste à contrôler les accés par salle et par horaires, en fonction d'un plan d'adressage de votre établissement. Le travail de l'académie de Grenoble avec ls projet "SLIS" permet de faire cela. On l'administre avec une interface Web. Ce n'est alors pas Squid qui est utilisé pour cela mais les fonctions de filtrage du routeur (netfilter par exemple). Construire des ACL directement dans Squid est faisable, mais cela n'est pas toujours simple à mettre en oeuvre.

La deuxième solution est de contrôler en fonction des individus. Squid permet de faire cela, à partir de plusieurs façons (APM, LDAP, NCSA auth, SMB...). Les différentes techniques sont décrites dans la FAQ de Squid sur le site officiel. *Squid* (http://www.squid-cache.org/related-software.html#auth)

Si vous utilisez un annuaire LDAP, vous devez avoir dans le fichier Squid.conf les lignes suivantes :

```
acl              identification   proxy_auth        REQUIRED
http_access      allow            identification
authentificate_program /usr/lib/squid/squid_ldap_auth \
                                     -b $LDAP_USER -u uid SERVEUR_LDAP
LDAP_USER est l'ou dans laquelle se trouve les clients
(par exemple ou=people, ou= ac-limoges, ou=education, ou=gouv, c=fr).
```

Si vous n'avez pas de serveur LDAP, une méthode simple à mettre en oeuvre, consiste à utiliser une méthode similaire ux fichier ".htaccess" d'Apache.

Exemple de configuration avec NCSA_auth

```
authenticate_program /usr/lib/ncsa_auth /etc/squid/passwd
acl                  foo                proxy_auth REQUIRED
acl                  all                src 0/0
http_access          allow              foo
http_access          deny               all
```

1.7. Interface web de Squid et produits complémentaires

squid dispose en standard de quelques outils, mais sinon vous pouvez utiliser webmin. Vous trouverez également, sur le site officiel de squid, une liste de produits supplémentaires pouvant être interfacés avec Squid.

1.8. La journalisation

Squid journalise les transactions dans un fichier access.log. Ce fichier donne les informations sur les requêtes qui ont transité par Squid. Le fichier cache.log informe sur l'état du serveur lors de son démarrage. Le fichier store.log informe sur les objets stockés dans le cache.

Les dates indiquées dans le fichiers access.log indique le temps en secondes depuis le 1 janvier 1970 (format epoch), ce qui n'est pas trés facile à lire. Un petit script en perl, permet de recoder les dates :

```
#! /usr/bin/perl -p
s/^\d+\.\d+/localtime $&/e;
```

1.9. Confi gurer les clients

Pour configurer les clients, on peut utiliser la configuration manuelle ou la configuration automatique avec des fichiers ".pac" ou des fihciers ".reg" que l'on place dans le script de connexion des clients.

Configuration manuelle des clients (http://slis.ac-creteil.fr/navigateurs.html)

Configuration automatique (http://www.ac-creteil.fr/reseaux/systemes/linux/outils-tcp-ip/proxy-pac.htm)

1.10. Forcer le passage par Squid (Proxy transparent)

Il existe plusieurs solutions:

Configurer votre navigateur avec le bon proxy ou en utilisant le fichier de configuration automatique et le rendre impossible à changer. Mais cela nécessite que vous contrôliez les clients ce qui n'est pas toujours le cas.

Intercepter les requêtes sur le port 80 du routeur pour les rediriger sur Squid.

Vous devez alors avoir dans votre fichier squid.conf :

```
# Configuration de traitement des requêtes du client
    httpd_accel_host virtual
    httpd_accel_port 80
    httpd_accel_with_proxy on
    httpd_accel_uses_host_header on
    httpd_accel_single_host off
```

Puis ajouter la règle pour netfilter de redirection des requêtes sur le port 80

```
iptables -t nat -F PREROUTING

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 \
                                            -j REDIRECT --to-port 8080
# Les clients peuvent envoyer leurs requêtes sur le port 80 du proxy
# Le service NAT du routeur les redirige sur le port 8080
```

1.11. Le redirecteur SquidGuard

Squid dispose d'une fonctionalité qui permet de passer une URL (requête entrante) à une applications externe. Cela présente l'avantage de pouvoir bénéficier des services d'applications spécialisées. C'est par exemple le cas pour le redirecteur SquidGuard, largement utilisé pour protéger les accès sur des sites déclarés comme "impropres". Une base de données de ces sites est tenue à jour. C'est cette dernière qui ets utilisée pour filtrer les accès.

1.12. Les applications non prises en charge par un service proxy

Certaines applications ne sont pas prises en charge par Squid (https, smtp, pop, ftp...). Les raisons peuvent être diverses. Soit le service n’est pas pris en charge (pop, smtp...), soit il n’est pas conseillé de stocker en cache certaines informations d’authentification par exemple (https).

Pour les applications ou services nons pris en charge par un service proxy, vous devrez utiliser l’ipmasquerade, un service de translation d’adresse ou utiliser une autre technologie.

2. Application

Vous devez maîtriser les techniques de routage avec netfilter.

Vous allez installer un service proxy minimal, configurer les clients puis tester le fonctionnement de l’accès à internet à partir des clients.

Vous configurerez des ACLs permettant un contrôle d’accès aux données externes, vous ferez ensuite évoluer cette configuration vers un service mandataire transparent.

Le service proxy sera installé sur le routeur.

Utilisez les éléments de ce document, ainsi que les exemples de fichiers de configuration donnés en annexe. Vous pourrez également vous référer au document sur netfilter.

2.1. Préparation de la maquette

Vous avez un routeur qui vous relie au réseau de l’établissement et un client qui représente un segment de réseau privé. L’ensemble doit fonctionner (accès à internet, résolution de nom, masquage d’adresse.

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -j MASQUERADE
# si 192.168.0.0 est le réseau privé
```

Vérifier que le routeur fonctionne. Faites un test à partir du client. Supprimez au besoin toutes les règles iptables et activez l’ipmasquerade.

Mettez une règle qui interdise toute requête à destination d’une application HTTP (port 80). Vérifier que les clients ne peuvent plus sortir.

```
# Ici on bloque tout, c’est brutal, mais on va faire avec.
iptables -P FORWARD DROP
```

2.2. Installation et confi guration du service proxy

Faites une sauvegarde de votre fichier de configuration original (/etc/squid.conf). Modifiez le fichier de configuration de squid en vous appuyant sur celui donné ci-dessous.

```
http_port 3128
#We recommend you to use the following two lines.
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 8 MBM

maximum_object_size_in_memory 8 KB

cache_dir ufs /var/spool/squid 100 16 256

cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log

# Put your FQDN here
visible_hostname freeduc-sup.foo.org

pid_filename /var/run/squid.pid

#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255

#Recommended minimum configuration:
http_access allow manager localhost
http_access allow all
```

Initialisez l'espace disque pour le cache.

```
# Vérifiez que le FQDN de votre serveur est renseigné
# et que la résolution de nom locale fonctionne (fichier hosts ou DNS).
# initialisation de la zone de cache
squid -z
# Lancement de squid
/etc/init.d/squid start | restart
```

Démarrer et vérifier le bon fonctionnement de Squid. Consultez également les journaux.

```
$>ps aux |  grep squid
root      2984  0.0  0.4  4048 1124 ?  S  15:22   0:00 \
                                              /usr/sbin/squid -D -sYC
proxy     2987  2.1  1.6  6148 4068 ?  S  15:22   0:00 (squid) -D -sYC
```

Vérifier que le port d'écoute est correct.

```
mlx@uranus:~$ netstat -atup | grep LISTEN
(Tous les processus ne peuvent être identifiés, les infos sur
```

```
les processus non possédés ne seront pas affichées, vous devez
être root pour les voir toutes.)

tcp        0      0 *:3128          *:*               LISTEN      -
```

Identifiez l'endroit de stockage du cache sur le disque.

2.3. Confi guration du client

Configurer le client pour qu'il utilise le service proxy sur les requêtes HTTP, vérifier le bon fonctionnement.

Figure 26. Configuration du client

Identifiez les traces dans les journaux.

```
root@uranus:/home/mlx# more /var/log/squid/access.log
1053864320.437   1741 192.168.0.2 TCP_MISS/200 5552 \
   GET http://www.cru.fr/documents/ - DIRECT/195.220.94.166 text/html
1053864320.837   1096 192.168.0.2 TCP_MISS/304 331 \
   GET http://www.cru.fr/styles/default.css - DIRECT/195.220.94.166 -
1053864321.257    420 192.168.0.2 TCP_MISS/304 331 \
   GET http://www.cru.fr/logos/logo-cru-150x53.gif - DIRECT/195.220.94
1053864321.587    696 192.168.0.2 TCP_MISS/304 331 \
   GET http://www.cru.fr/icons-cru/mailto.gif - DIRECT/195.220.94.166
1053864550.537   1461 192.168.0.2 TCP_MISS/200 5552 \
   GET http://www.cru.fr/documents/ - DIRECT/195.220.94.166 text/htm
```

Interdisez tous les accès avec la règle :

```
http_access deny all
```

Vérifiez le fonctionnement.

2.4. Mise en place d'une ACL simple

Interdisez l'accès à un serveur (google.fr) par exemple. Vérifiez le fonctionnement.

```
acl google dstdomain .google.fr
http_access deny google
```

2.5. Utilisation de fichiers pour stocker les règles des ACL

Construisez deux fichiers, l'un qui permettra de stocker des adresses IP, l'autres des mots clés. Construisez une ACL qui interdit l'accès en sortie aux machines qui ont les adresses IP déterminées dans le premier fichier, et une ACL qui empêche l'accès aux URL qui contiennent les mots clés stockés dans le second fichier.

```
# Exemple de ce que le fichier "adresse_ip" contient :
# Mettez dans laliste des adresse celle de votre client pour tester
192.168.0.2
192.168.0.10

# Exemple de ce que le fichier "mot_cle" contient :
jeu
game

# Exemple d'ACL
acl porn url_regex "/etc/squid/mot_cle"
acl salleTP_PAS_OK src "/etc/squid/adresse_ip"
http_access deny porn
http_access deny salleTP_PAS_OK
```

Tester le fonctionnement de ces deux ACL. (Utiliser comme url de destination par exemple : http://games.yahoo.com/)

2.6. Configuration des messages d'erreurs

Configurez Squid pour qu'il affiche des pages (messages d'erreur) en Français. Vérifiez le fonctionnement.

```
error_directory /usr/share/squid/errors/French
```

Identifiez la page qui est retournée lors d'un refus d'accès. Modifiez la page et le message retourné, puis vérifiez le fonctionnement.

2.7. Automatisation de la configuration des clients.

Créez un fichier ".pac" pour la configuration des clients Mozilla. Vous en avez un complet dans la FAQ de squid. Celui-ci fait le minimum.

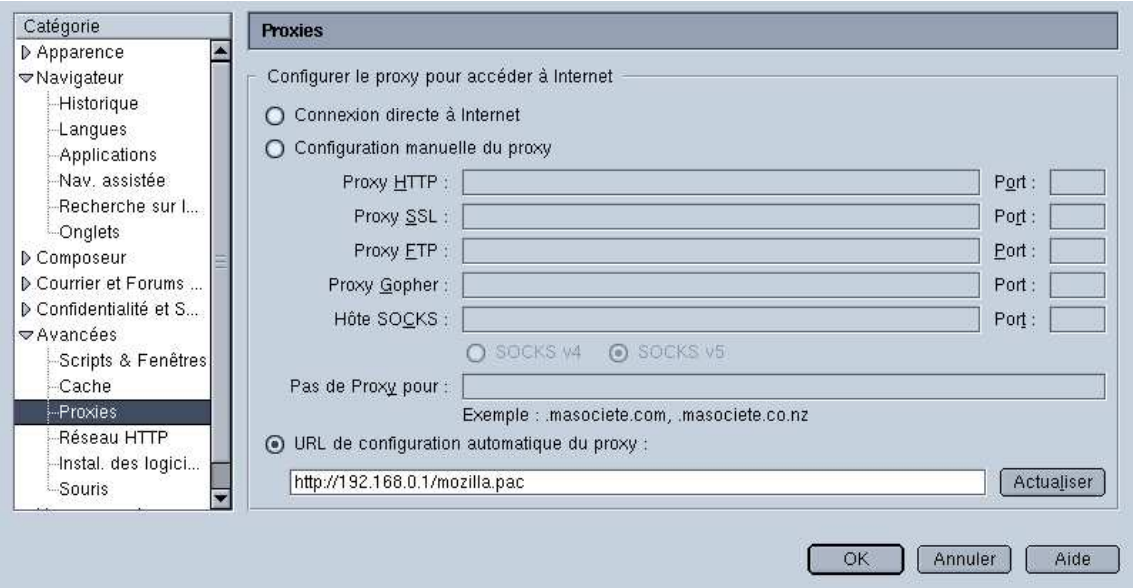
```
 function FindProxyForURL(url, host)
   {
```

```
 return "PROXY 192.168.0.1:3128; DIRECT";
}
```

Mettez le fichier sur votre routeur dans /var/www/mozilla.pac et vérifiez que le serveur apache est bien démarré. Si la résolution de nom fonctionne, vous pouvez mettre le nom du serveur de configuration plutôt que l'adresse IP.

Configurez le client :

Figure 27. Configuration du client



Testez le bon fonctionnement du client.

Remettez la configuration du client dans sa situation initiale.

2.8. Installation et confi guration du service proxy Squid transparent.

Modifier la configuration du client et du serveur, afin que la configuration globale devienne celle d'un proxy tranparent.

Vous allez modifier le fichier de configuration de squid et configurer votre routeur avec les règles suivantes si le service proxy est sur le routeur :

```
# A mettre dans le fichier de configuration de suid
# Relancer le service après
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy  on
httpd_accel_uses_host_header on
```

```
# Les règles iptables
# On nettoie la table nat
# On utilise le port 3128, par utilisé par défaut sous Squid.
iptables -t nat -F PREROUTING

# ou toutes les tables nat si besoin
iptables -t nat -F

# On laisse passer (masque) les requêtes autres que sur le port 80
 iptables -t nat -A POSTROUTING -j MASQUERADE

# On redirige les requêtes sur le port 80
iptables -t nat -A PREROUTING -j DNAT -i eth1 -p TCP --dport 80 \
                                    --to-destination 192.168.0.1:3128
```

Supprimer toute configuration de proxy sur le client. Vérifier le bon fonctionnement du client.

Arrêtez les service proxy, vérifiez que les requêtes HTTP des clients ne sortent plus.

Il est possible de séparer les services du routage et proxy sur 2 machines différentes. Le principe est identique, seules les règles sur le routeur changent un peu. Vous trouverez la description d'une telle configuration dans :

```
# Transparent proxy with Linux and Squid mini HOWTO
http://www.tldp.org/HOWTO/mini/TransparentProxy.html

# Lire aussi sur netfilter
http://www.netfilter.org/documentation/HOWTO/fr/NAT-HOWTO.txt
http://www.cgsecurity.org/Articles/netfilter.html
```

2.9. Mise en place de l'authentification

Mettez en place une ACL pour déclarer l'authentification des personnes.

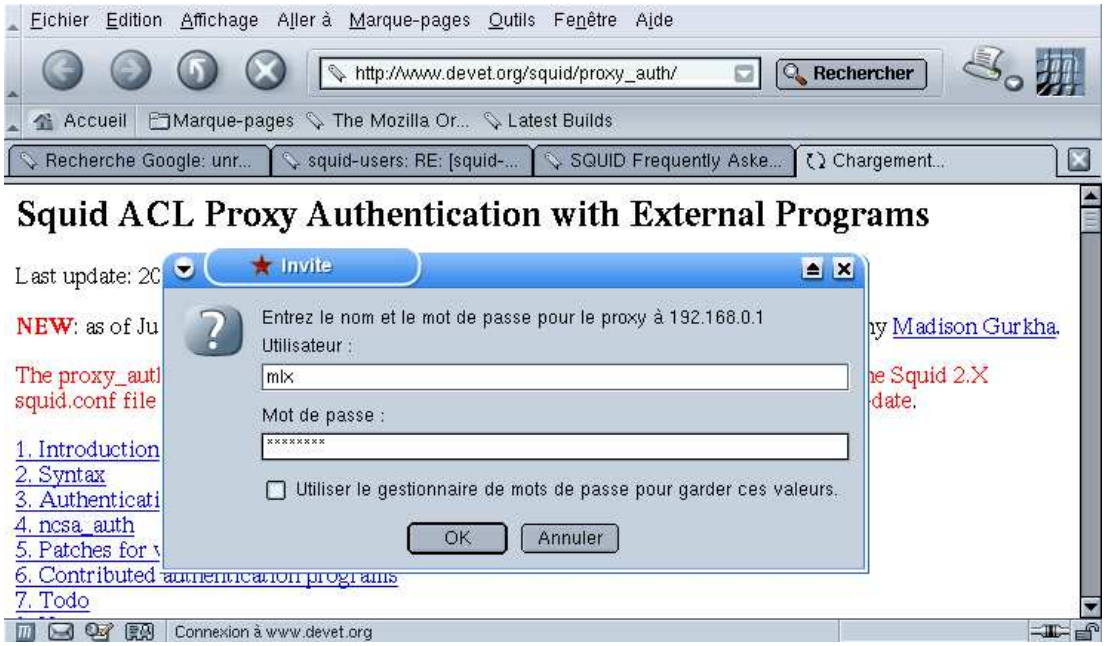
```
# Ici on utilise le module ncsa_auth
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/users
auth_param basic realm Squid proxy-caching web serve
auth_param basic children 5
acl foo proxy_auth REQUIRED
http_access allow foo
```

Créez les fichiers de compte et de mots de passe avec un compte utilisateur.

```
 htpasswd -c /etc/squid/users unUTILISATEUR
# mettez ensuite son mot de passe.

# Testez le fonctionnement du fichier et du module
# Vous passez en paramètre le nom du fichier de comptes
# Vous mettez le compte et le mot de passe, le module retourne OK
```

```
# En cas d'erreur il retourne ERR
root@uranus:/etc# /usr/lib/squid/ncsa_auth /etc/squid/users
mlx password
OK
mlx mauvais
ERR
```

Figure 28. Authentification SQUID



Il y a pas mal de différences entre les paramètres des versions de Squid 1, squid 2 et Squid 2.5. Il est important de consulter les fichiers de documentation fournis avec le produit.

L'authentification ne fonctionne pas avec la configuration d'un proxy transparent.

3. Liens

1. *Squid* (http://www.squid-cache.org/)
2. *Le CRU* (http://www.cru.fr/renater-cache/cache_regional.html)
3. *SquidGuard - Université de Toulouse* (http://cri.univ-tlse1.fr/documentations/cache/squidguard.html)
4. *Les HOWTOs* (http://www.freenix.fr/unix/linux/)

4. Annexes

4.1. Fichier squid.conf - testé avec Squid 2.5

Fichier minimal pour Squid

```
http_port 3128

#Ne pas "cacher" les données des formulaires
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 8 MBM

maximum_object_size_in_memory 8 KB

cache_dir ufs /var/spool/squid 100 16 256

cache_access_log /var/log/squid/access.log
cache_log /var/log/squid/cache.log
cache_store_log /var/log/squid/store.log

# Ici mettez le nom de votre machine
visible_hostname uranus.freeduc-sup.org

pid_filename /var/run/squid.pid

#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255

# Test des fichiers @ip et mots clés
acl porn url_regex "/etc/squid/mot_cle"
acl salleTP_PAS_OK src "/etc/squid/adresse_ip"
http_access deny porn
http_access deny salleTP_PAS_OK

# Authentification
auth_param basic program /usr/lib/squid/ncsa_auth /etc/squid/users
auth_param basic realm Squid proxy-caching web serve
auth_param basic children 5
acl foo proxy_auth REQUIRED
http_access allow foo

#Default:
#http_access deny all

#Messages d’erreurs en FR
error_directory /usr/share/squid/errors/French

# Pour le proxy cache transparent
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy  on
httpd_accel_uses_host_header on
```

4.2. Exemples d’ACLs Squid 2.2

Utiliser des fichiers externes pour la déclarations d’adresse ou de mots clés.

```
acl salleTP_OK src "/etc/squid/salleTP_OK.txt"
acl porn url_regex "/etc/squid/porn.txt"
acl salleTP_PAS_OK src "/etc/squid/salleTP_PAS_OK.txt"
http access salleTP_OK
http access porn
http deny    salleTP_PAS_OK
```

4.3. ACL par authentifi cation Squid 2.2

Utilisation d’une authentification simple similaire à celle mise en oeuvre dans les ".htaccess". Créer le fichier par script ou manuellement avec htpasswd.

```
authenticate_program /usr/bin/ncsa_auth /etc/squid/users
authenticate_children 5
acl_authenticate_users REQUIRED
http_access authenticate_users
```

4.4. ACL sur des plages horaires Squid 2.2

Combinaison par "ET" logique des plage horaire et des salles. Mettre la machine à l’heure avec ntpdate par exemple.

```
# Interdire les accès en dehors des plages horaires 8h-12h et 14h-18h
S Sunday M Monday T Tuesday W Wednesday H Thrusday F Friday A Saturday
acl am time MTWHF 08:00-12:00
acl PM time MTWHF 14:00-18:00
http_access allow am salleTP_PAS_OK
http_access allow pm salleTP_PAS_OK
```

Installation d’un serveur PostgreSQL avec Apache

Création d’un site web dynamique avec PostgreSQL et Apache. Pour PostgreSQL vous pouvez aussi utiliser la ressource Linux-France.org (http://www.linux-france.org/prj/edu/archinet/BD/index/) qui détaille de façon assez complète un mode d’utilisation de ce serveur de bases de données.

1. Avant de démarrer

Si vous utilisez la Freeduc-Sup rc3, un bogue empêche PostgreSQL de se lancer. Vous devez avoir un message d’erreur dans "/var/log/postgres" qui vous indique qu’il n’arrive pas à trouver un fichier "pg_control".

Voici comment corriger cela : sous le compte root taper

```
mv /var/lib/postgres/data /var/lig/postgres/data.old
dpkg-reconfigure postgresql
OK
OK
Yes
OK
fr_FR@euro
OK
LATIN1
OK
ISO
European
OK
NO
#C’est terminé, vous pourrez lancer PostgreSQL normalement.
```

Cela sera corrigé sur la prochaine version.

2. Les ressources sur PostgreSQL

Vous avez un support de cours, TD et TP assez complet sur Linux-France (http://www.linux-france.org/prj/edu/archinet)qui décrit bien le mode d’utilisation de PostgreSQL.

3. Accès aux archives

Vous pourrez récupérer les documents nécessaires sous forme d’archive sur le serveur de linux-france. Pour cela voir la page d’introduction du document.

4. Présentation

Accès à une base de données PostgreSQL à partir d’un client WEB (Mozilla ou autres)

On veut à partir d’un client "Web" comme Mozilla (ou autres) interroger une base de données PostgreSQL. Le client HTTP passe (via des formulaires) des requêtes SQL à un serveur Web sous Linux (Apache). Celui-ci dispose d’une interface "PHP" qui lui permet d’interroger la base de données. En fait Apache va "lancer" l’exécution de "scripts PHP" et éventuellement récupérer et retourner les résultats d’exécution au client.

Les processus mis en jeu côté serveur sont les suivants :

HTTPD qui va permettre les accès via le Web (Gestion des formulaires)

Postmaster qui est le daemon gérant tous les accès à la base.

Le serveur disposera également des documents HTML et des scripts PHP

- Le travail à réaliser en TP consistera donc à :
- - Créer une base de données Postgres
- - Démarrer le daemon postmaster permettant sa gestion
- - Démarrer le daemon HTTPD
- - Accéder à la base de données (via httpd) à l’aide de scripts PHP.

5. Présentation de PostgreSQL

PostgreSQL est un système de gestion de base de données, développé à l’origine par l’université de Berkeley. Il s’appuie sur les modèles relationnels mais apporte des extensions objet comme :

- les classes,
- l’héritage,
- les types de données utilisateurs (tableaux, structures, listes..),
- les fonctions,

- supporte complètement SQL,
- portable sur plus de 20 environnements depuis la version 6.4.

Cela permet de qualifier PostgreSQL de système de gestion de base de données "relationnel-objet" (ORDBMS), à ne pas confondre avec les bases de données orientées objets qui ne supportent pas SQL, mais OQL (Object Query Language).

PostgreSQL est diffusé avec ses sources (licence libre).

5.1. Mode de fonctionnement de PostgreSQL

Les trois composantes majeures sont :

- un processus de supervision (daemon) qui prend en charge les connexions des clients : postmaster,
- les applications clientes comme psql, qui permettent de passer des requêtes SQL,
- le ou les serveurs de bases de données (*agents*). Processus d'ouverture de session : (voir le schéma d'ouverture de session.)

5.1.1. Description du processus d'ouverture de session

1. Le client passe une requête au daemon *postmaster* via un socket. Par défaut sur le *port 5432*. La requête contient le nom de l'utilisateur, le nom de la base de données. Le daemon, peut à ce moment utiliser une procédure d'authentification de l'utilisateur. Pour cela il utilise le catalogue de la base de données, dans lequel sont définis les utilisateurs.
2. Le daemon crée un alors un *agent* pour le client. Le processus serveur répond favorablement ou non en cas d'échec du démarrage du processus. (exemple : nom de base de données invalide).
3. Le processus client se connecte sur le processus agent. Quand le client veut clore la session, il transmet un paquet approprié au processus agent et ferme la connexion sans attendre la réponse.
4. Plusieurs processus *agents* peuvent être initialisés pour un même client.

5.1.2. Le dictionnaire :

Comme pour la plupart des systèmes de gestion de données, toutes les informations système sont stockées dans des tables qui forment le dictionnaire (catalogue ou repository en Anglais). Utiliser le cataloque est essentiel pour les administrateurs et les développeurs. Vous pouvez voir la structure et le contenu de ces tables système.

5.1.3. PostgreSQL fournit :

un langage d’administration (création de base, d’utilisateurs)

un langage d’interrogation de données basé conforme à SQL

des extensions C, C++, perl, php, python...

5.1.4. Les comptes utilisateurs :

Le compte administrateur de la base est par défaut " *postgres* "

il faut créer les comptes utilisateurs

Voir le TP sur HTTP pour obtenir le compte système qui est utilisé par Apache pour les requêtes http. Sur la freeduc-sup c’est "www-data". Dans la suite du document on utilisera $COMPTE_HTTP pour parler de ce compte système.

5.2. Langage de commande pour PostgreSQL

Voici quelques commandes d’administration de base :

Création d’une base de données : createdb

createdb [dbname]

createdb [-h host] [-p port] [-D datadir] [-u] [dbname]

Exemple : createdb -h uranus -p 5432 -D PGDATA -u demo

ou encore *createdb demo*

Suppression d’une base de données

dropdb [dbname]

Exemple *dropdb demo*

Créer un utilisateur :

createuser [username]

createuser [-h host] [-p port] [-i userid] [-d | -D] [-u | -U] [username]

-d | -D permet ou interdit la création de base à l'utilisateur

-u | -U permet ou interdit la création d'autres comptes à l'utilisateur.

Crée un compte dans pg_user ou pg_shadow. (tables système)

Si la base est accessible par Internet (exemple avec PHP), l'accès est réalisé par le compte " $COMPTE_HTTP ".

Utiliser la commande "select * from pg_user;" pour avoir la liste des utilisateurs.

Supprimer un utilisateur

drop user [username]

Accéder à une base:

psql [dbname]

psql -A [-c query] [-d dbname] -e [-f filename] [-F separator] [-h hostname] [-o filename] [-p port] -qsSt]

[-T table_options] -ux [dbname]

```
mlx@mr:~$ psql template1
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=#
```

6. Présentation de PHP

PHP, (Personal Home Page) est un langage de programmation complet, assez proche du C. Il fournit :

- des structures de données,

- des structures de contrôle,
- des instructions de gestion des entrées/sorties.

Il est diffusé également sous licence libre. Il permet la création de pages web dynamiques.

Il est considéré comme une alternative à CGI, Perl, ASP (Active Server Page de Microsoft);

Développé à l'origine pour Linux, il est maintenant portable sur plusieurs environnements (Windows 9.x, NT).

Il fournit des API pour les bases de données Oracle, PostgreSQL, MySQL, DB2 ;, et est conforme aux standards ODBC et ISAPI

Il fonctionne avec de nombreux serveurs HTTP comme Apache ou IIS (Internet Information Server) de MS.

PHP peut être utilisé seul ou combiné avec des bases de données et un serveur HTTP (Objet du TP).

Simple à mettre en oeuvre, documenté, sécurisé et fiable, de nombreux sites (FAI) comme libertysurf, free mettent cet outil à la disposition des clients.

6.1. Mode de fonctionnement de PHP

Sur Linux, PHP est compilé comme un module dynamique ou directement intégré à Apache, ce qui accroît les performances.

Le code PHP peut être intégré directement dans une page HTML comme vb-script ou à l'extérieur sous forme de fonctions (comme CGI).

Le code est logé entre deux balises < ? Ici le code ?>. Il est possible que pour assurer la compatibilité avec XML, les balises deviennent : <php et ?>

L'extension généralement utilisée pour les documents PHP est .php. Voir ci-dessous l'exemple " test.php " qui permet de vérifier le support de PHP par votre environnement.

Listing : test.php

```
<?
echo (" Test du module PHP ");
phpinfo();
?>
```

6.2. Le langage PHP

Le guide utilisateur et ses extensions comprennent plus de 300 pages (voir les sources de documentations plus bas). La description ci-dessous donne les principales instructions pour les accès à une base de données PostgreSQL.

pg_Connect : Connexion à une base de données :

int pg_connect(string host, string port, string options, string tty, string dbname);

Retourne faux si la connexion échoue, un index dans l’autre cas. Il peut y avoir plusieurs connexions.

Exemple : $conn = $conn = pg_Connect("localhost", "5432","","", "template1");

Ou : $conn = pg_connect("dbname=marliese port=5432");

pg_Close : Fermer une connexion

bool pg_close(int connection);

pg_cmdTuples : Donne le nombre de tuples affectés par une commande insert,

update ou delete. Renvoie 0 sinon.

int pg_cmdtuples(int result_id);

Exemple :

<?php

$result = pg_exec($conn, "INSERT INTO verlag VALUES (’Autor’)");

$cmdtuples = pg_cmdtuples($result);

echo $cmdtuples . " affectés.";

?>

string pg_dbname(int connection);

Donne le nom de la base de données.

Exemple $NomBase = pg_Dbaname ($conn);

pg_ErrorMessage :

string pg_errormessage(int connection);

Message d'erreur renvoyé par le serveur

pg_Exec : int pg_exec(int connection, string query);

Exécute une requête.

$UneChaineSQL = "Select * from UneTable");

Exemple : $result = pg_exec($conn, $UneChaineSQL);

pg_FieldName : string pg_fieldname(int result_id, int field_number);

Renvoie le nom du champ d'indice field_number ;

Exemple :

indice = 0

While (indice [lt] NombreDeChamp)

{

$NomChamp = pg_fieldname($result, indice)

echo $NomChamp

indice ++;

}

pg_FieldNum : int pg_fieldnum(int result_id, string field_name);

Donne l'indice pour un nom de champ.

pg_Host : string pg_host(int connection_id);

Donne le nom du Host

pg_NumFields : int pg_numfields(int result_id);

Renvoie le nombre de champs de la requête.

Exemple : $numF = pg_Numfields($result);

pg_NumRows : int pg_numrows(int result_id);

Renvoie le nombre de tuples (enregistrements) de la requête.

Exemple : $numR = pg_NumRows ($result);

if ($numR == 0)

{

echo "Aucun enregistrement retourné. ";

exit;

}

pg_Result : mixed pg_result(int result_id, int row_number, mixed fieldname);

Renvoie la valeur d'un champ, pour un n° d'enregistrement donné et un résultat de requête. Les numéros d'enregistrement et de champ commencent à 0.

Exemple avec $i - indice d'enregistrement et $j - indice de champ :

$Valeur = pg_result ($conn, $i, $j)

pg_Options : pg_Options (int connection_id);

Renvoie une chaîne contenant les options de connexion à la base.

pg_FreeResult : int pg_freeresult(int result_id);

Libérer la mémoire.

Autres fonctions de base :

pg_Fetch_Array, pg_Fetch_Object, pg_Fetch_Row, pg_FieldsNull, pg_PrtLen,

pg_FieldSize, pg_FieldType, pg_GetLastOid, pg_port, pg_tty.

Vous trouverez la documentation de ces commandes dans celle de PHP.

7. Dialogue client et serveurs PHP, Apache et PostgreSQL

- Une requête SQL est passée par un formulaire HTML ou autre et via le protocole HTTP
- Le serveur Apache reçoit la requête HTTP
- Le module PHP exécute la requête sur la base PostgreSQL en utilisant les API
- Le code PHP met en forme le résultat de la requête
- La page est remise au serveur Apache
- Le serveur Apache retourne le résultat au client.

Vous avez deux méthodes pour passer les paramètres au serveur : la méthode "GET" et al méthode "POST".

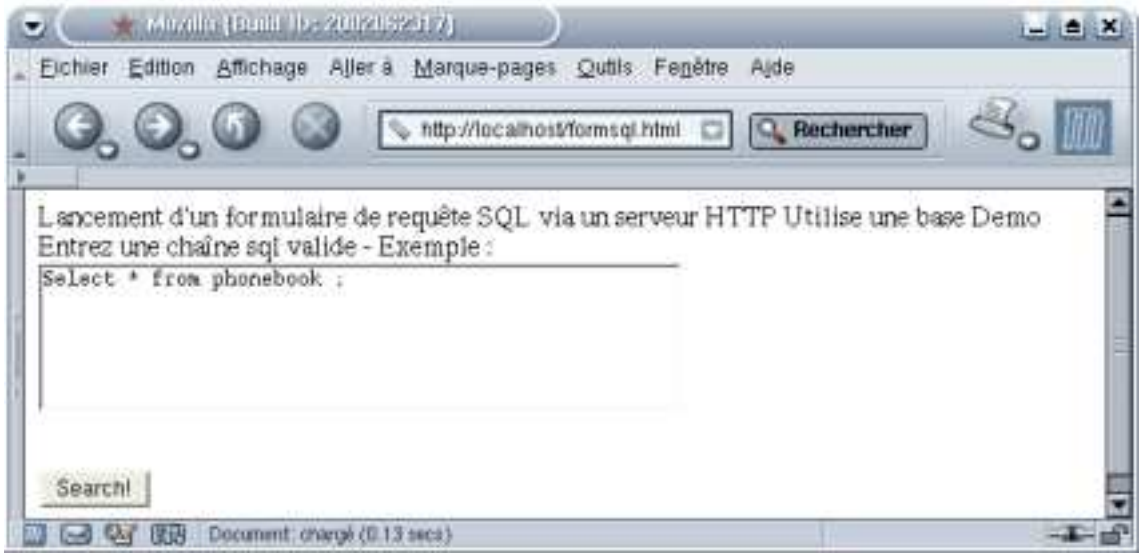
8. Exemple de code

Voici un exemple de formulaire html et le script PHP associé.

Le formulaire : formsql.html

```
<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
<head>
</head>
<body>
Lancement d’un formulaire de requête SQL via un serveur HTTP
Utilise une base Demo
<br>
Entrez une chaîne sql valide - Exemple :
<form action="resultsql.php" METHOD=post> // Ici le script qui sera exécuté
<textarea cols="50" rows="5"
name="c_SQL">Select * from phonebook ;</textarea></p>
<br>
<INPUT TYPE="submit" VALUE="Search!">
</form>
</body>
</html>
```

Figure 29. Formulaire de saisie

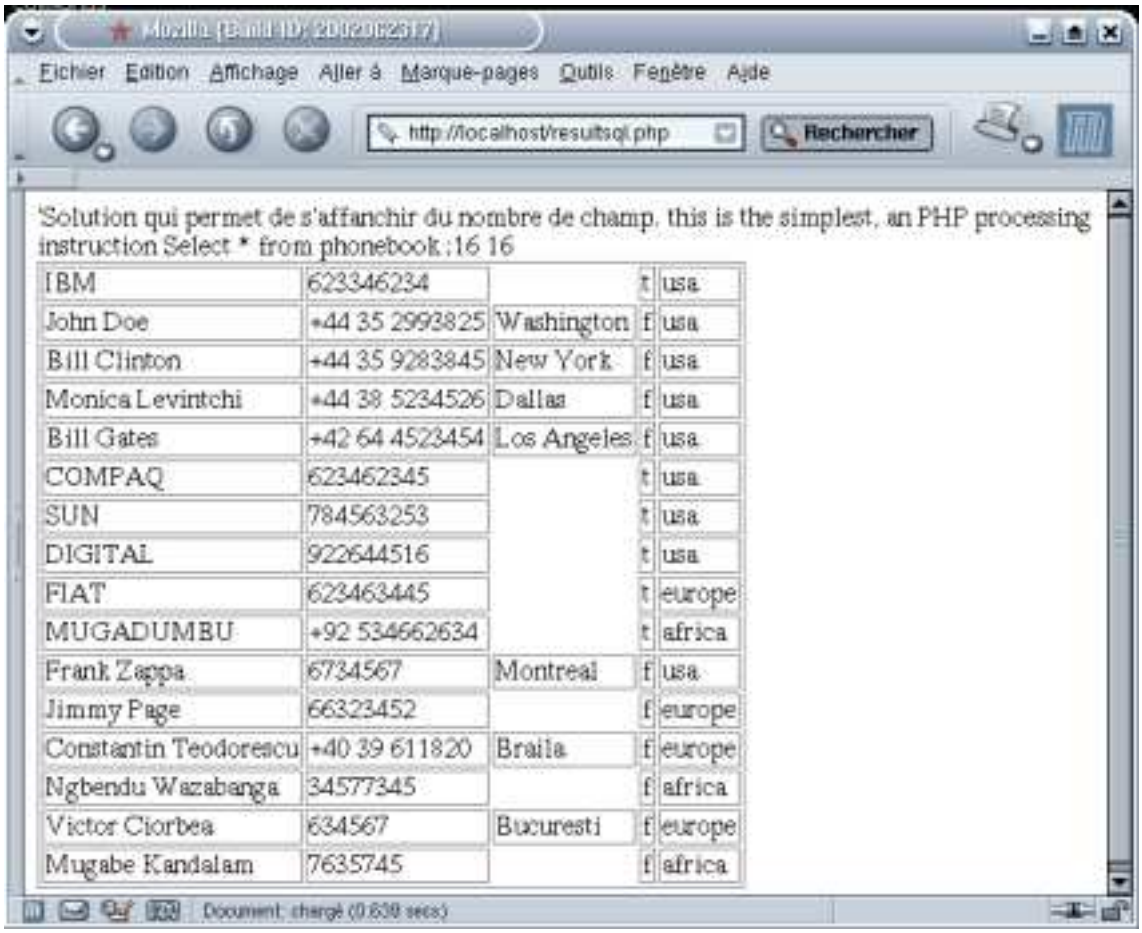


Le script associé : Page resultsql.php

```
 'Solution qui permet de s'affranchir du nombre de champs.
 <?
/* Test de la connexion à la base */
 if($c_SQL != "")
 {
 echo $c_SQL ;
 $conn = pg_Connect("localhost","5432","","","demo");
  if (! $conn)
 {
  echo "Erreur de connection à la base. \n";
  exit;
  }

 /* teste le résultat de la requête */
 $result = pg_Exec($conn, $c_SQL);
  if (! $result)
 {
  echo "Erreur d'accès aux tables. \n";
  exit;
  }

/* teste le nombre de tuples retournées */
  $numR = pg_NumRows ($result);
  if ($numR == 0)
 {
  echo "Aucun enregistrement retourné. \n";
  exit;
  }

 /* Compte le nombre de champs */
 $numF = pg_Numfields($result);
```

```
 /* mise en forme du résultat sous forme tabulaire */
 /* lignes (tuples), colonnes (champ) */
 echo "<table border = 1>";
 $i = 0;
 while ($i < $numR)    {
   echo "<tr>";
  $j = 0;
    while ($j < $numF)  {
   $nc=pg_result($result,$i,$j);
   echo "<td>"; echo $nc;  echo "</td>";  $j++;

   }
  echo "</tr> \n";
  $i++;
  }
 echo "</table> \n";
/* Libère la mémoire */
 pg_FreeResult;
/* Ferme la connection */
 pg_Close($conn);
 }
 ?>
```

Figure 30. Résultat de la requête



9. TP

9.1. Présentation

Accès à une base de données PostgreSQL à partir d’un serveur Apache. Utilisation du langage PHP.

La maquette terminée devrait permettre, à partir d’un client HTTP comme Netscape de passer des requêtes SQL à un serveur Apache. Le serveur Apache dispose d’une interface PHP, qui lui permet d’échanger avec une base de données PostgreSQL .

Vous devrez récupérer pour le TP les documents suivants :

1. Le script de création de la base de démo "formdemo.sql"
2. le document HTML "formsql.html"

3. le document php "resultsql.php"

9.2. PostgreSQL

Connectez-vous en tant que *root.*

1 Préparation de la configuration

Installez les packages correspondant à PostgreSQL et à PHP s’ils ne sont pas déjà installés.

2 Configuration de Postgres

2.1 Postgres est installé. Le script de lancement est dans *"/etc/init.d"*

Editez ce script et relevez :

l’emplacement où sont stockées les bases de données.

Recherchez le port et les protocoles de transports utilisés par PostgreSQL avec la commande *"grep postgres /etc/services"*

Vérifiez que le fichier "/etc/postgresql/postmaster.conf" comporte bien la ligne "POSTMASTER_OPTIONS="-i -p 5432". Cela permet de définir le numéro de port, et d’inquer à PostgreSQL de supporter les sessions sockets (-i).

Vous allez configurer les options de sécurité permettant au serveur de recevoir des requêtes. Ouverz le fichier "/var/lib/postgres/data/pg_hba.conf". Recherchez les lignes ci-dessous :

```
# Put your actual configuration here
# ----------------------------------
host         all         127.0.0.1     255.0.0.0           ident sameuser
host         all         0.0.0.0       0.0.0.0             reject
```

Modifiez ces lignes après avoir fait une copie de sauvegarde de ce fichier, de la façon suivante :

```
host         all         127.0.0.1     255.0.0.0           trust
host         all         x.y.z.t       x’.y’.z’.t’         trust
```

Vous adapterez "x.y.z.t" à l’adresse de votre réseau et "x’.y’.z’.t’" au masque de votre réseau.

2.2 Il s’agit maintenant d’activer le service. Utilisez les commandes :

/etc/init.d/postgresql stop

/etc/init.d/postgresql start

Vérifiez le chargement de postgres dans la table des processus : *"ps aux | grep post"*

Vérifiez dans les journaux les messages d’erreurs si le serveur ne démarrer pas.

Vérifiez également :

- qu’ un service n’est pas déjà actif,

- que les variables sont bien déclarées. En général les messages de Postgres sont assez clairs et donnent la marche à suivre pour corriger. N’allez pas plus loin tant que tout cela ne fonctionne pas parfaitement.

3 Tester la configuration

La procédure précédente à créé un modèle de base de données "template1", qui sert de modèle pour la création d’autres bases, et a créé un compte d’administrateur de base de données "*Postgres*". Toujours en mode commande et en tant qu’utilisateur postgres (*su postgres*), vous allez utiliser la commande suivante :

psql template1

Attention, il n’y a qu’un seul compte de base de données, celui de l’administrateur "postgres". Vous allez ouvir uen session sous le comtpe "root" puis passer sous le compte "postgres" avec la commande "su postgres".

Vous devriez obtenir ceci :

```
# su postgres
sh-2.05b$ psql template1
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=#
```

Le caractère "=>" est le prompt du mode commande. Vous pouvez désormais taper des commandes.

Retenez "\q" pour quitter.

Pour avoir de l'aide sur l'interpréteur de postgres : *template1=> \?*

Pour avoir de l'aide sur les commandes SQL : *template1=> \h*

Si l'aide s'affiche, c'est que tout fonctionne. Par contre vous ne pouvez pas faire grand chose, la base et vide. Vous pouvez le vérifier avec la commande "\dt".

template1=>\dt

Couldn't find any tables!

template1=\q

Tout autre message, signifie qu'il y a un problème de configuration. Si c'est le cas vérifiez soigneusement tous les paramètres.

4 Conclusion

Votre environnement fonctionne et est bien configuré. La prochaine étape consiste à se familiariser avec les premières commandes d'administration et d'utilisation.

9.3. Test de la base

1 Créer une base de données

1 - Création de la base :

Vous devez avoir récupéré le script de création de la base "formdemo.sql"

su postgres (Vous devez être administrateur de la base)

createdb demo (création d'une base de données s'appelant *demo*)

2 - Création des tables de la base *demo* :

psql demo < formdemo.sql

2 Test de la base de données

Vous allez, au préalable, tester le fonctionnement de tout cela à partir du compte Administrateur *"postgres"*. Pour cela utilisez les commandes suivantes:

psql demo

Pour afficher les tables

=> \dt

consultez la table *phonebook*. Vous devriez avoir le résultat.

=>*select * from phonebook;* (Ne pas oublier *;*)

#quitter

=> \q

3 Créer un compte d'utilisateur de base de données

Vous allez créer et utiliser deux comptes utilisateurs de bases de données. "*$COMPTE_HTTP*" qui est utilisé pour les accès HTTP, "*TP1* " que vous utiliserez comme compte local. Vous leur affecterez pour l'instant les droits minimums.

3.1 Normalement le compte système*$COMPTE_HTTP* existe déjà, vous pouvez vérifier avec *"grep $COMPTE_HTTP /etc/passwd"*. Si ce n'est pas le cas, vous devrez créer un compte système pour $COMPTE_HTTP.

3.2 Création du compte système "TP1"

Création du compte

adduser TP1

affectation d'un mot de passe.

passwd TP1

3.3 Vous allez créer les comptes de base de données pour $COMPTE_HTTP et TP1. Attention aux réponses que vous mettrez car $COMPTE_HTTP ne doit pas avoir la possibilité de créer des tables, ni créer d'autres comptes de bases de données.

3.3.1 Création du compte anonyme $COMPTE_HTTP

#passer en DBA (Data Base Administrator)

su postgres

$ *createuser $COMPTE_HTTP*

Enter user's postgres ID or RETURN to use unix user ID: 99 ->

Is user "$COMPTE_HTTP" allowed to create databases (y/n) *n*

Is user "$COMPTE_HTTP" allowed to add users? (y/n) *n*

createuser: $COMPTE_HTTP was successfully added

#c'est terminé, voilà le résultat :

```
demo=# \q
sh-2.05b$ createuser www-data
Shall the new user be allowed to create databases? (y/n) n
Shall the new user be allowed to create more new users? (y/n) n
CREATE USER
sh-2.05b$
```

3.3.2 Création d'un compte DBA TP1

#passer en DBA (Data Base Administrator)

su postgres

$*createuser TP1*

Enter user's postgres ID or RETURN to use unix user ID: 501 ->

Is user "TP1" allowed to create databases (y/n) *y*

Is user "TP1" allowed to add users? (y/n) *y*

createuser: TP1 was successfully added

#c'est terminé

3.3.3 $COMPTE_HTTP n'a aucune permission sur les bases de données. Vous allez lui donner la permission de faire des "select".

Utilisez les commandes suivantes :

psql demo

grant select on phonebook to $COMPTE_HTTP ;

```
demo=# grant select on phonebook to "www-data";
GRANT
demo=#
```

\q

4 Tester l'accès des comptes

Ouvrez une session avec le compte *TP1* que vous avez créé.

psql demo

Pour afficher les tables

=> \dt

consultez la table *phonebook*. Vous devriez avoir le résultat.

*=>select * from phonebook;* (Ne pas oublier le *;*)

#quitter

=> \q

9.4. Serveur Apache et PHP

Démarrez le serveur Apache : */etc/init.d/apache restart*

Vérifiez que le serveur est bien actif et opérationnel.

Cherchez et relevez l'emplacement de stockage (Home Directory) des pages html d'Apache.

Vérification de la prise en charge de php par apache :

Normalement il n’y a plus rien à faire. Il s’agit de vérifier que le module PHP est bien pris en charge par Apache. Voici comment procéder:

1 - Créer dans Home Directory d’Apache le document *testphp.php* suivant:

```
<? echo ("Test du module PHP");
phpinfo();
?>
```

2 - Lancez un navigateur (à partir de votre poste ou d’une autre machine) et tapez l’url "*http:///@IP de votre PC/testphp.php* » .

Deux solutions :

- soit le résultat est bon, la fonction "phpinfo()" vous retourne des informations sur le module et sur Apache. Dans ce cas vous pourrez continuer,

Figure 31. Interrogation de PHP

- soit ce n’est pas le cas, il faut revoir la configuration.

9.5. Serveur PostgreSQL/Apache et PHP

Le serveur HTTP et le client fonctionnent, php est pris en charge par le serveur Apache. Maintenant, nous allons créer un formulaire qui permet de passer des requêtes SQL sur la base et un script qui exécute ces requêtes.

1 Test de la demo

1.1 En fait il s'agit de deux documents (*formsql.html* et *resultsql.php*) :

- le premier est une page HTML qui permet de saisir et "passer" des requêtes SQL,
- le deuxième au format PHP, met en forme le résultat de la requête.

1.2 Installez les pages fournies dans le répertoire d'Apache.

Lancez ensuite un navigateur. Tapez l'url http://@IP du PC/formsql.html. Vous pouvez saisir une chaîne sql et "envoyer" le formulaire.

2 Modifications

a) Copier *insert.html* dans le répertoire d'Apache, modifiez les permissions du compte $COMPTE_HTTP afin de lui donner la possibilité d'insérer des tuples dans la base de données. Modifiez le document resultsql.php afin de pouvoir réaliser des insertions dans la base. Les enregistrements à insérer seront saisies dans insert.html.

Figure 32. Formulaire insert.html

9.6. TP de synthèse

`Durée de réalisation 20h en binômes`

Vous allez créer une base de données conforme au schéma relationnel suivant :

```
cours (cou_no, cou_lib)
etudiant (etu_no, etunom)
incrit(cou_no, etu_no)
```

On demande de développer l'interface web qui permet de :

1. ajouter nouveau cours
2. supprimer un cours
3. modifier le libellé d'un cours
4. ajouter un étudiant à un cours