Table des matières

Liste des illustrations

Installation d’un intranet avec des logiciels libres

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. 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.

Préparation des machines

Alix MASCRET

Mode d'utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Résumé

Ce document décrit une procédure pour installer, dans une salle de cours, les environnements Windows 98, Windows NT et Linux sur des machines à vocation pédagogique.

Mots clés : « NTLDR », « LILO », « Dual Boot »

1.1. Objectif

Un poste de travail dans une salle de TP, en BTS Informatique de GESTION, nécessite le plus souvent l'installation d'au moins deux systèmes d'exploitation Windows 9x et Windows NT. Avec Linux, cela va en faire un troisième. Windows NT dans sa version station ou sa version serveur, prend en charge l'initialisation de plusieurs systèmes. Comme Linux fait la même chose, la question sera de savoir, quel système de NT ou Linux, sera chargé d'assurer cela.

Ce document donne quelques indications pour répondre à la question suivante :

Comment configurer une machine avec les trois systèmes d'exploitation (Windows 98, NT, Linux) dans une salle de TP ?

Il faut avoir des notions assez précises sur les partitions systèmes, partitions d'amorçage, partitions étendues, master boot record (MBR) et partition boot record (PBR).

1.2. Configuration matérielle

Il faut que le matériel soit dans la HCL pour NT et compatible avec Linux.

Prévoir 32 à 64 MO de RAM

- Windows 9x 200 Mo
- NTServer 200 Mo
- Linux 700 Mo

1.2.1. Création du fichier BOOTSEC.LIN

- démarrez la machine sous Linux,
- vérifiez que la partition de Windows est bien montée,
- tapez la commande dd if=/dev/hda3 bs=512 count=1 of=/dos/bootsec.lin

Commentaire : on lit en entrée le premier secteur de 512 octets sur /dev/hda3. La sortie est redirigée dans le fichier bootsec.lin, localisé sur /dos (/dev/hda1).

Relancez la machine pour démarrer sous NTS ou Windows 9x.

Adaptez la commande à votre configuration.

1.2.2. Modification du fichier BOOT.INI

- modifiez les attributs du fichier car il est en lecture uniquement et caché,
- ouvrez-le à l'aide d'un éditeur,
- rajoutez à la fin la ligne *c:\bootsec.lin="Linux Mandrake"*
- enregistrez,

Si tout fonctionne correctement, remettez les attributs d'origine sur le fichier BOOT.INI.

2. TP

Les machines disposent d'un disque de 9 Go et 128 Mo de RAM. Tous les composants sont reconnus par Linux. Préparez votre disque avec 4 partitions primaires. Vous répartirez le même volume pour chacun des systèmes d'exploitation (W 9x, W2K ou NT4 Server, Linux). Vous garderez 128 Mo pour la partition de swap.

Préparez l'installation des machines afin que vous puissiez installer :

- Windows 9x sur la première partition
- Linux sur la deuxième partition
- W2K ou NT4 Server sur la troisième partition
- Une partition de swap de 128 Mo.

Vous commencerez par Linux. Prévoyez de faire votre disquette de démarrage. Les autres systèmes seront installés ultérieurement.

Installation de Telnet et FTP

Alix MASCRET

Mode d'utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Résumé

Le document décrit l'installation d'un service FTP, la configuration du démon inetd.

Mots clés « Telnet », « FTP », « 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 TCPWrapper.

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 complètement identique, 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.

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

telnet stream tcp nowait root /usr/sbin/telnetd telnetd

#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 telnet qui est activé par le serveur inetd. Les autres lignes sont en commentaires.

Ces services sont dits fonctionnant en mode « parallèle ».

1.5. Configuration 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. TCPWrapper

5) TCPWrapper:

Tcpwrapper 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 configuration

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:

```
/etc/hosts.deny   /etc/hosts.allow
in.ftpd:ALL   in.ftpd :cli1.archinet.edu
```

interdit tous les accès ftp à la machine, autorise les accès ftp venant de cli1.

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 :

décrire toutes les règles pour les couples (services/clients) qui sont autorisés,

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!
*.info;mail.none;authpriv.none; /var/log/messages

# The authpriv file has restricted access.
authpriv.* /var/log/secure
```

1.9. Extrait de /var/log/messages

```
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
uid=0)
Feb 3 18:10:57 ns1 login[1063]: LOGIN ON ttyp3 BY mlx FROM puce
```

Remarques:

La commande Kill -HUP pid de syslogd permet de redémarrer ce processus avec prise en compte des paramètres se trouvant dans /etc/syslog.conf.

1.10. Processus d'installation

La procédure est assez simple. Si TCP-Wrapper est installé, vérifier que les services sont installés (ls -al /usr/bin/in*).

La procédure d'installation crée au moins 2 répertoires dans /home dont les répertoires :

/home/ftp/pub /* répertoire public pour le service ftp anonyme */

/home/ftp/incoming /* répertoire en écriture pouvant recevoir les dépôts */

D'autres répertoires (lib, etc, bin) peuvent être créés. Ces répertoires servent à « chrooter » le daemon ftp. Cette procédure n'est pas abordée ici.

Il n'y a plus rien à faire, sauf si vous voulez sécuriser le service avec TCPWrapper.

Procédure de configuration

Ouvrez le fichier /etc/inetd.conf, vérifiez que la ligne qui active le démon ftp est décommentée.

Pour l'activer manuellement utilisez la commande : /etc/rc.d/init.d/inet 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 « ftp localhost » ou « ftp 'hostname' » en vous authentifiant

ftp anonyme « ftp localhost » ou « ftp 'hostname' » en utilisant le compte anonymous

« telnet localhost » ou « telnet 'hostname' »

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 « inetd.conf »

1 - décommentez la ligne qui contient le chargement de ftp,

2 - vérifiez les messages dans /var/log/messages et /var/log/secure

3 - relancez le serveur /etc/rc.d/init.d/inet stop ; /etc/rc.d/init.d/inet start

Vous devriez voir les demandes ftp rejetées par TCPWrapper..

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. TP

- 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/rc.d/init.d/NomDuService start | stop | status | restart"

3. 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
```

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:
```

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

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

Résolution de noms d’hôtes avec un fichier HOST

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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’hôtes 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 adresse IP en adresse 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 ?

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

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. 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.

Vous verrez également comment sécuriser des accès à certaines pages, ainsi que la façon de mettre en place des pages dynamiques à l’aide de la technique des CGI.

2. 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 qui seront vus plus tard. Vous allez réaliser les TP(s) de cet atelier en mode commande.

2.1. Installation du package principal: rpm -i apache*.rpm

La procédure installe:

- le binaire httpd dans /usr/sbin,
- les fichiers de configuration dans /etc/httpd/conf (httpd.conf, et des modules complémentaires pour Postgres, Mysql par exemple)
- la documentation dans le répertoire racine du serveur HTTP, qui est dans/home/httpd ou dans /var/www en fonction des distributions
- Le script de lancement du service serveur dans /etc/rc.d/init.d

Faites une copie de sauvegarde des fichiers de configuration.

2.2. 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 /etc/httpd/conf/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 nobody et group nobody, 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.
- 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/httpd, indique l'adresse du répertoire racine du serveur. Cette adresse peut être modifiée.

- ErrorLog logs/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 /etc/httpd/conf/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 /home/httpd/html, 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/ /home/httpd/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/, 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/

- ScriptAlias est identique au paramètre alias, mais pour l'exécution de scripts.
- 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,

- 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>
  #Dans l'ordre on interdit tout accès, ensuite on gère les autorisations
  order deny, allow
  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 site.

  ```
  <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 2 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.

  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:

2.3. Activation du serveur

Utilisez les commandes suivantes pour activer, désactiver ou voir l'état du serveur:

/etc/rc.d/init.d/httpd start

/etc/rc.d/init.d/httpd stop

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

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

/etc/rc.d/init.d/httpd reload

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

2.4. Test de la configuration

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 aux autres machines de la salle, et également à celles d'Internet si votre machine est configurée pour cela.

3. 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

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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 Netscape.

La résolution de nom doit fonctionner.

La description est faite avec une distribution Linux de RedHat. L’emplacement des fichiers change avec d’autres distributions. Vous devrez en tenir compte.

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, en général /homt/httpd/html ou /var/www/html
- $APACHE_CONF, répertoire dans lequel sont stockés les fichiers de configuration, en général /etc/httpd/conf
- $APACHE_USER, compte utilisteur utilisant Apache, en général nobody
- $APACHE_GROUP, groupe auquel est rattaché le compte nobody, en général nobody.

2. TP1 - Installation d’un serveur Web

2.1. Introduction

Vous allez réaliser les opérations suivantes:

- installez le package du serveur Apache,
- 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 Toutes les commandes ne sont pas indiquées, vous devrez rechercher et noter celles qui manquent.

2.2. Première partie : Installation des packages

Désinstallez Apache s’il est installé par défaut avec l’installation. Utilisez rpm -qa pour voir la liste des packages installés et rpm -e pour désinstaller

Montez le cdrom et installez Apache. Une fois cette opération réalisée, il ne reste plus qu’à le configurer.

2.3. Deuxième partie : 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. Attention, en fonction des versions et des distributions certains paramètres par défaut peuvent changer. Pour chacun de ces paramètres vous noterez leurs rôles à partir des commentaires donnés par les fichiers httpd.conf. (pensez à enregistrer vos modifications):

- ServerType standalone
- Port 80

- User $APACHE_USER
- Group $APACHE_GROUP
- ServerAdmin root@localhost
- ServerRoot /etc/httpd
- 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/

2.4. Troisième partie : Activation du serveur

Regardez dans la fiche de cours les commandes de lancement du service serveur et de la procédure de test. Regardez dans les fichiers de log et dans la table de processus si le service est bien démarré.

Notez toutes les commandes que vous utilisez.

2.5. Quatrième partie : 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.html.

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.

2.6. 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,
- Installez puis déployez un site Web,
- Développez quelques pages HTML puis les déployer,
- Testez les nouvelles pages à partir d’un client Linux et Windows.

3.2. Première partie - 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.

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. Troisième partie - 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/html/journal pour y mettre toutes les pages html
- Copiez les images dans $APACHE_HOME/icons
- Copiez le script cgi dans $APACHE_HOME/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 et l’exécution du script.

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

3.4. Quatrième partie - 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/html/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/html/cv

3.5. Cinquième partie - Test de vos pages

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

Essayez d’expliquer pourquoi vous arrivez à vous connecter au serveur HTTP, indifféremment depuis les environnement Linux et Windows alors que les systèmes d’exploitation sont complètement différents ?

3.6. Sixième partie - 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.
- Pouvez vous apporter une conclusion sur cette manipulation ?

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.
- Pourquoi un navigateur sous Windows arrive-t-il à lire des documents situés sur un serveur Unix ?
- Pourquoi un navigateur sous Windows arrive-t-il à lancer des scripts compilés pour une machine Unix ?
- Quel intérêt procure l’utilisation des alias ?
- Qu’est ce qu’un lien Hypertexte et comment cela fonctionne-t-il ?
- 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. Première partie - Configurer 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. Deuxième partie - 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 et le script CGI de votre répertoire personnel "/home/mlx/public-html/cgi-bin" et pas ceux qui sont dans "$APACHE_HOME/cgi-bin"

4.3. Troisième partie - 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:

  - l’adresse URL de son site personnel,

  - l’emplacement physique de son répertoire personnel sur la machine,

  - 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. Première partie - Déployer un site d’accès en ligne

Vous allez utiliser les pages fournies en annexe (Le journal du monocle - html.zip). *Pages HTML* (images/html.zip).

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

Copiez les pages dans ce répertoire.

5.2. Deuxième partie - 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/html/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/html/protege>
AuthName Protected
AuthType basic
AuthUserFile $APACHE_CONF/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. Troisième partie - Tester la configuration.

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 avec les machines des autres étudiants et faites tester votre configuration.

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/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.

5.4. Auto-évaluation sur le quatrième TP

- Dans quel cas un accès sécurisé peut-il être utilisé ?
- On désire limiter l’accès d’un répertoire uniquement aux personnes qui appartiennent au domaine. A l’aide des exemples ci-dessous, dites quelle est la bonne stratégie à utiliser:

  #Première solution

  order deny, allow

  deny from all

  order from MonDomaine

  #Deuxième solution

  order allow, deny

  allow from MonDomaine,

  deny from all

- 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 ?
- Quelle autre solution peut-on utiliser pour sécuriser un répertoire ?
- Pourquoi faut-il refermer le navigateur avant chaque batterie de test ?

6. TP5 - Utilisation de scripts CGI

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:

- 1 - Etudiez les sources fournies en annexe, (package cgi.zip) *Sources C CGI* (images/cgi.zip).
- 2 - Développez un formulaire et adaptez les scripts,
- 3 - Testez le fonctionnement des scripts.

6.1. Première partie - Etudier les sources fournies en annexe

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

6.2. Deuxième partie - Développer un formulaire et adapter les scripts

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 :

- Liste déroulante (Calzone, Margarita, Quatre-saisons)
- Bouton radio (Grande, Moyenne, Petite)
- Vous afficherez au client le résultat de sa commande.

Adaptez le script CGI pour que la commande soit enregistrée dans un fichier de type texte. Les nouvelles commandes seront ajoutées en fin de fichier.

6.3. Troisième partie - Tester le fonctionnement des scripts.

Pour tester le script:

- ouvrez la page principale de votre site à l'aide d'un navigateur,
- passez des commandes,
- vérifiez les résultats de la page html (réponse) puis le contenu du fichier résultat.

Faites tester votre site par d'autres étudiants à partir de machines distantes, testez également leurs sites.

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 ?

Installation d’un serveur SAMBA

Le partage de fichier pour les clients Windows

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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 configuration de SAMBA

2.1. Installer le produit

Installez la version de samba livrée avec votre distribution. Il ne devrait normalement pas y avoir de problèmes de dépendances.

La procédure installe principalement :

- les programmes nmdb et smdb dans /usr/bin,
- le script d’initialisation dans /etc/rc.d/init.d,
- un fichier de configuration /etc/smb.conf,
- une documentation assez importante dans /usr/doc.

La procédure crée également un dossier pour samba dans /home.

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

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 ‘hostname‘ comme contrôleur de domaine,
- l’authentification des utilisateurs,
- le partage des disques et une imprimante pour un client Windows,
- le partage 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 :

- Installer le fichier de configuration,
- Déclarer les ressources partagées,
- Créer un compte d’utilisateur autorisé.

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

2.4. Première étape - Installer le fichier de configuration

Copiez le fichier texte "smb.conf" dans le répertoire /etc et activez le service avec la commande "/etc/rc.d/init.d/smb start | stop | status | restart". 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 par le fichier smb.conf. Chaque fois que vous ajoutez ou modifiez une ressource relancez le service serveur.

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 des commandes «smpaddusers » et « smbpasswd ». Vous trouverez ces scripts dans /usr/bin.

Par exemple : smbadduser mlx:ntidmlx où ntidmlx est le ntid (NT Identifier) de l’utilisateur mlx. Cela crée 2 fichiers /etc/smbusers et /etc/smbpasswd si ces fichiers n’existent pas. Modifiez le mot de passe de l’utilisateur mlx à l’aide de la commande « smbpasswd mlx », c’est terminé.

Trois remarques :

- Les manipulations peuvent paraître fastidieuses si vous avez un grand nombre de comptes utilisateurs.

- Si vous disposez de nombreux comptes d’utilisateur sur votre système Linux, vous disposez d’un script qui permet de créer automatiquement les comptes Samba à partir du fichier /etc/passwd. Vous aurez deux manipulations si vous utilisez les fichiers "ombres" où "shadow passwd".
- Vous pourrez synchroniser la base des comptes utilisateurs Linux avec la base des comptes Samba.

Vous pouvez créer des comptes utilisateurs Samba, sans que le compte utilisateur Linux existe.

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 NT de votre domaine. Vérifier la configuration du protocole TCP/IP, relancez la machine. Vous pourrez vous authentifier sur le serveur Samba.

Toutes les commandes net use.., 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, faites attention aux droits, car si "root" est propriétaire (chmod 700), le client ne pourra pas accéder au disque.

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

```
Annexe 1 : Exemple de fichier de configuration de Samba :

#====================== Fichier SMB.conf==========================

# Fichier de configuration de samba
# Permet le partage de dossier en lecture seule en lecture / écriture
# des imprimantes

#====================== Global Settings ============================
```

```
[global]

# ATTENTION : Penser à :
# créer un répertoire dans /home/netlogon
# accessible au compte nobody (chgrp nobody)
# modifier le "guest account" en nobody ou alors créer
# un compte spécifique. Les clients
# Windows ont besoin de ce compte d'accès pour les services
# d'exploration et pour l'inscription auprès du serveur Wins.

# workgroup = NT-Domain-Name or Workgroup-Name
workgroup = VotreDomaine

# Déclaration du nom Netbios et de ses aliases
netbios name = Nom_NetBIOS
netbios aliases = d2r2 pluton cristal

# server string is the equivalent of the NT Description field
# %v permet d'afficher le N° de version de Samba dans l'explorateur
# du client Windows
# %h permet d'afficher le nom d'hôte du serveur Samba
server string = Samba Server version %v - host %h

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page

# Adaptez les permissions d'accès à vos réseaux
hosts allow = 192.168.1. 192.168.2. 127.


# Uncomment this if you want a guest account, you must
# add this to /etc/passwd otherwise the user "nobody" is used
# guest account = pcguest
# Le compte « nobody » existe déjà pour Apache ou d'autres services.
# Vous pouvez l'utiliser pour l'exemple mais jamais pour Samba si votre
# serveur est accessible d'internet. guest account = nobody

# this tells Samba to use a separate log file for each machine
# that connects
log file = /var/log/samba/log.%m

# Put a capping on the size of the log files (in Kb).
```

```
max log size = 50

# Security mode. Most people will want user level security. See
# security_level.txt for details.
# Si vous utilisez le mode « Server » décommentez le paragraphe qui suit.
security = user

# Use password server option only with security = server
; password server = <NT-Server-Name>

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
#Nécessaire pour les mots de passe cryptés
encrypt passwords = yes
# update encrypted = yes
smb passwd file = /etc/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux sytsem password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
# the encrypted SMB passwords. They allow the Unix password
# to be kept in sync with the SMB password.
; unix password sync = Yes
; passwd program = /usr/bin/passwd %u
; passwd chat = *New*UNIX*password* %n\n *ReType*new*UNIX*password* %n\n
# *passwd:*all*authentication*tokens*updated*successfully*

# Unix users can map to different SMB User names
; username map = /etc/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
# of the machine that is connecting
; include = /etc/smb.conf.%m

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
# Le serveur Samba va tenter de devenir (élection) maître explorateur.
```

```
local master = yes

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
# Le serveur va jouer le rôle de serveur NT
#NT Server utilise 32, NT Workstation utilise 16, Windows 95 utilise 1
# La valeur 33 assure au serveur Samba d'être prépondérant
# sur les autres machines et autres OS.
os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
# Si le Serveur est explorateur maître il élabore les listes d'exploration
# des clients du domaine et des groupes de travail.
domain master = yes

# Preferred Master causes Samba to force a local browser election on startup
# and gives it a slightly higher chance of winning the election
#Le serveur Samba est le serveur d'exploration privilégié
# du groupe de travail.
preferred master = yes

# Use only if you have an NT server on your network that has been
# configured at install time to be a primary domain controller.
; domain controller = <NT-Domain-Controller-SMBName>
# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
#Assure l'authentification des utilisateurs
domain logons = yes

# if you enable domain logons then you may want a per-machine or
# per user logon script
# run a specific logon batch file per workstation (machine)
# Permet l'exécution de scripts
# Préferrer des scripts c, perl aux batch standards avec les instructions
# preexec et postexec, %m utilise le nom de la machine
# Utiliser %U.bat pour utiliser les scripts existants sous nt et sous
# le nom %username%.bat
; logon script = %m.bat

# run a specific logon batch file per username
; logon script = %U.bat

# Where to store roving profiles (only for Win95 and WinNT)
```

```
# %L substitutes for this servers netbios name, %U is username
# You must uncomment the [Profiles] share below
; logon path = \\%L\Profiles\%U

# Windows Internet Name Serving Support Section:
# WINS Support - Tells the NMBD component of Samba to enable it's WINS Server
# Support WINS
wins support = yes

# WINS Server - Tells the NMBD components of Samba to be a WINS Client
# Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
# Déclarer un serveur WINS sur le réseau. Attention, cette option est
# exclusive avec l'option précédente.
; wins server = w.x.y.z

#====================== Share Definitions ======================

# Les répertoires personnels
[homes]
comment = Home Directories
browseable = no
writable = yes

# Un-comment the following and create the netlogon directory for
# Domain Logons.  A mettre impérativement si vous voulez utiliser
# Samba comme serveur d'authentification.
# Pensez à créer le répertoire
[netlogon]
comment = Network Logon Service
path = /home/netlogon
guest ok = yes
writable = no
share modes = no

# Un-comment the following to provide a specific roving profile share
# the default is to use the user's home directory
# Permet d'utiliser les profils utilisateurs et profils errants.
;[Profiles]
; path = /home/profiles
; browseable = no
; guest ok = yes

# NOTE: If you have a BSD-style print system there is no need to
# specifically define each individual printer
#Permet de partager une imprimante Linux par Samba pour des clients Windows
```

```
[printers]
comment = All Printers
path = /var/spool/samba
browseable = no
# Set public = yes to allow user 'guest account' to print
guest ok = no
writable = no
printable = yes

# This one is useful for people to share files
# Un répertoire partagé. Utiliser le sticky bit pour simplifier
#l'administration.
[tmp]
comment = Temporary file space
path = /home/tmp
read only = no
writable = yes
public = yes

# Un autre répertoire de partage autre que celui de /tmp
[partage]
comment = Temporary file space
path = /home/tmp
read only = yes
public = yes

# A publicly accessible directory, but read only, except for people in
# the "staff" group
# Dossier accessible en lecture uniquement sauf pour les membres
# du groupe « staff »
[ronly]
comment = Public Stuff
path = /home/samba
public = yes
writable = yes
printable = no
write list = @staff
```

4. TP

4.1. 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 NT.

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.

4.2. Déroulement des opérations

- Les opérations vont se dérouler en 6 étapes :
- 1 - Installation du fichier RPM de Samba sur le serveur,
- 2 - Configuration du fichier /etc/smb.conf et démarrage des services,
- 3 - Création d'un compte utilisateur,
- 4 - Création du fichier d'authentification pour Samba /etc/smbpasswd,
- 5 - Création de ressources disques partagées en lecture et en lecture/écriture,
- 6 - Configuration du client Windows 9x,
- 7 - Procédure de test.

4.3. Installation du fichier RPM de Samba sur le serveur

Installez les packages si cela n'est pas déjà réalisé.

4.4. Configuration du fichier /etc/smb.conf et démarrage des services

Le fichier de configuration smb.conf est dans le répertoire /etc. 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.

Voici un exemple avec quelques commentaires. Etudiez-le sérieusement et remplacez les paramètres à modifier.

```
 [global]
# workgroup = NT-Domain-Name or Workgroup-Name
 workgroup = XXXXXXXXXXXXXXXXXX
# server string is the equivalent of the NT Description field
 server string = XXXXXXXXXXXXXXXXXXXX
# this tells Samba to use a separate log file for each machine that connects
 log file = /var/log/samba/log.%m
# Put a capping on the size of the log files (in Kb).
 max log size = 50
# Security mode. Most people will want user level security. See
# security_level.txt for details.
 security = user
# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# On utilise les mots de passe cryptés avec le fichier /etc/smbpasswd
encrypt passwords = yes
;smb passwd file = /etc/smbpasswd
# Unix users can map to different SMB User names
; username map = /etc/smbusers
# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
socket options = TCP_NODELAY
# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
 os level = 33
# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
#Le serveur servira à l'authentification,
#il sera contôleur de domaine et sera maître explorateur.
domain logons = yes
local master = yes
domaine master = yes
preferred master = yes
#====================== Share Definitions ======================
```

```
# Dossier personnel
[homes]
comment = Home Directories
;browseable = no
writable = yes
# Dossier partagé
[tmp]
comment = Temporary file space
path = /tmp
read only = no
public = yes
```

Pour arrêter ou démarrer les services :/etc/rc.d/init.d/smb start | stop | restart | status

4.5. 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 des commandes « smbadduser » et « smbpasswd » pour modifier le mot de passe. Utilisez la commande "smbadduser help" pour en connaître le mode d’utilisation. Le paramètre « ntid » est un identifiant système/windows. Il doit être unique. Vous pouvez utiliser comme ntid la concaténation des chaînes « ntid »
« VotreCompteUtilisateur », exemple « ntidlupo », si lupo est le compte que vous avez créé.

4.6. 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) ouvrez-le avec un éditeur et réalisez la procédure de test qui y est décrite.

4.7. Procédure de test à partir d'un client windows

Configurez votre client Windows pour qu'il puisse faire partie de votre domaine NT (Panneau de configuration, icône Réseau).

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.

4.8. Procédure de désinstallation.

Désinstallez sur votre PC sous linux le serveur Samba.

Vous devrez avant cela vider le répertoire /var/log.samba.

Effacez le fichier /etc/smb.conf.

Supprimez sur le serveur les comptes utilisateurs créés.

Remettez le client Windows dans son état original.

Installation d’un serveur DHCP

Alix MASCRET

Mode d’utilisation du serveur DHCP Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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 Provider (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 Provider 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

3. Indication pour la réalisation du TP

Le processus va se dérouler en 4 étapes:

- l'installation du binaire
- la configuration du serveur
- la configuration des clients
- le test de la configuration.

3.1. Installation du serveur

Installation du serveur

Installez les packages dhcp à partir du cdrom. Cette procédure va installer les binaires serveur et client ainsi que le script de lancement du serveur.

3.2. Configuration du serveur

La configuration consiste à créer 2 fichiers:

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

- */etc/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 /etc/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.

```
[root@uranus /etc]# more dhcpd.conf

# ici il s'agit du réseau 192.168.1.0
subnet 192.168.1.0 netmask 255.255.255.0{

#La plage d'adresse disponible pour les clients
range 192.168.1.10 192.168.1.20;
```

```
# Les clients auront cette adresse comme passerelle par défaut
option routers    192.168.1.254;

# Ici c'est le serveur de nom, on peut en mettre plusieurs
option domain-name-servers   192.168.1.1;

# Enfin on leur donne le nom du domaine
  option domain-name   "planete.net";

# Et l'adresse utilisée pour la diffusion
option broadcast-address 192.168.1.255;

#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 192.168.1.125;
 }
host m2 {
 hardware ethernet a0:81:24:a8:e8:3b;
 fixed-address 192.168.1.126;
 }
    }#End Group
}#End dhcp.conf
```

3.2.2. Création d'un fichier d'inscription /etc/dhcpd.leases

Ce fichier doit être créé, sans quoi le serveur DHCP ne pourra pas démarrer. Il suffit de créer un fichier vide. Pour cela taper la commande *cat /dev/null > /etc/dhcpd.leases.* Le fichier est créé. Voici ce que l'on peut avoir dedans après l'inscription du premier client:

```
[root@uranus /etc]# more  /etc/dhcpd.leases
lease 192.168.1.10 {
 starts 1 1998/12/14 18:33:45;
ends 1 1998/12/14 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 le service: */etc/rc.d/init.d/dhcpd stop*

- pour activer le service : */etc/rc.d/init.d/dhcpd start*

Voici ce que donne la commande d'activation:

```
[root@uranus /etc]# /etc/rc.d/init.d/dhcpd start
Starting dhcpd: dhcpd Internet Software Consortium
DHCPD $Name: V2-BETA-1-PATCH LEVEL-6 $
Copyright 1995, 1996, 1997, 1998 The Internet Software Consortium.
All rights reserved.
Listening on Socket/eth0/192.168.1.0
Sending on   Socket/eth0/192.168.1.0
```

3.3. Installation des clients

3.3.1. Le client sous Windows 9x

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/sysconfig/network-script. C’est ici qu’est la configuration des cartes installées sur la machine. Ouvrez le fichier *ifcfg-eth0* à l’aide d’un éditeur et vérifiez les paramètres suivants:

```
Device=eth0    #interface utilisée, ici il n’y a qu’une carte
Userctl=no     #l’utilisateur ne peut modifier les paramètres de la carte
Onboot=yes     #active au démarrage du système
Bootproto=dhcp #boot protocole utilisé DHCP
Brodcast=      #tous les autres paramètres sont distribués par le serveur DHCP
Network=
Netmask=
Ipaddr=
```

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

3.4. Procédure de test

Sur Windows 9x vous allez pouvoir utiliser 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.
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.

Installation d’un serveur DNS - TD

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Résumé

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

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 Linux.

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

Le service de résolution de nomss 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 l’exemple de cet atelier il s’agit de "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 nomss 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,
- 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 egalement en TP le fonctionnement de la réplication des bases de données (bases d’enregistrements de ressources.

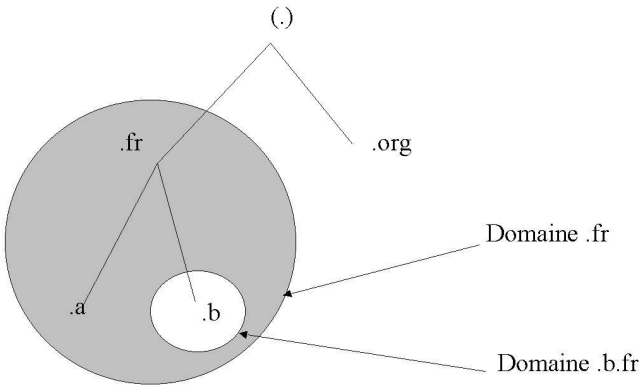
4. Présentation des concepts

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 hiérachique 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.

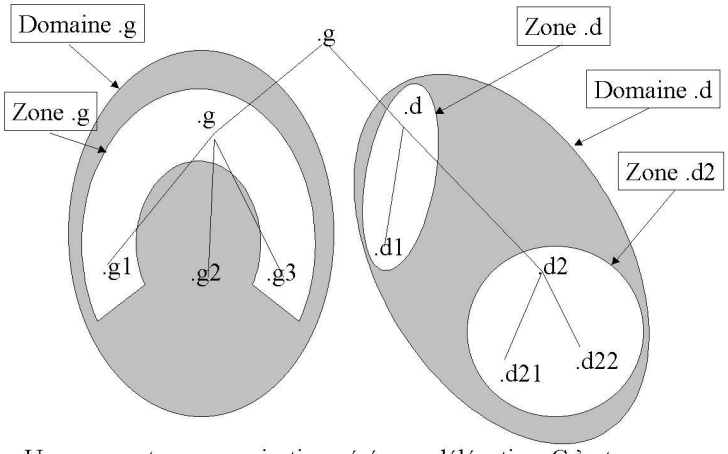
Figure 1. Les domaines



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.

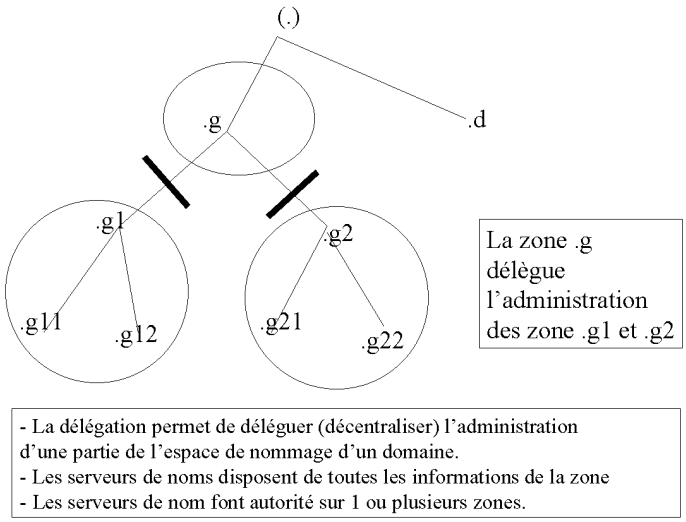
Figure 2. Les zones



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 3. La délégation



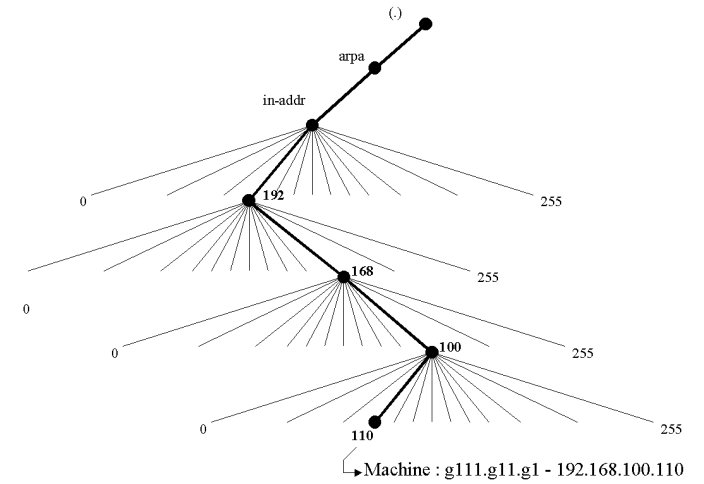
Attention à ces quelques remarques :

- 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.
- Les seules machines connues au niveau de l'espace de nommage, sont les serveurs de nom "déclarés".
- La cohérence (le service de résolution de noms) entre l'organisation de l'espace de nommage et les organisations physiques des réseaux sur internet et réalisées par les serveurs de noms.

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 4. La résolution inverse



L’adresse ip 192.168.100.110 correspond au sous-domaine 110.100.168.192.in-addr.arpa qui renvoie le nom qualifié g111.g11.g1

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)

4.3. Fichiers, structure et contenus

Sur linux nous allons utiliser deux types de fichiers :

- le fichier /etc/named.conf, qui décrit la configuration générale du serveur DNS,
- les fichiers qui contiennent les enregistrements de ressources pour la zone dans /var/named.

Les enregistrements ont une structure et un rôle.

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.
- Enregistrements de type CNAME (Canonical Name) : Ils permettent de définir des alias sur des noeuds existants.
- 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.

4.5. Structure des enregistrements

Strucure d’un enregistrement SOA : Chaque fichier commence par un enregistrement de type SOA. Voici un exemple d’enregistrement SOA.

```
foo.org. IN SOA ns1.foo.org. hostmaster.foo.org. (
```

```
20001210011     ; numéro de série
10800           ; rafaîchissement
3600            ; nouvel essai
604800          ; Obsoléscence 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.

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

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 période de test de la validité de ses données.
3. Retray : Intervalle de temps avant réïtération 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.
foo.org. IN NS ns2.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.

```
ns1.foo.org. IN CNAME www.foo.org.
ns2.foo.org. IN CNAME ftp.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.
```

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.

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.

4.8. Le cache

L’organisation d’internet est assez hiérarchique. Chaque domaine dispose de son propre serveur de noms. Chaque zone de niveau supérieur (edu, org, fr...) dispose également de serveurs de nom de niveau supérieur. Le service DNS installe une liste de serveurs de noms de niveaux supérieurs. Cette liste permet à votre serveur de résoudre les noms qui sont extérieurs à votre 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).

5. Installation d’un serveur DNS

Processus d’installation

Pour mettre en place le service de résolution de noms sur un serveur Linux, on va procéder successivement aux opérations suivantes :

1. installer le package si cela n’est pas déjà réalisé,
2. configurer les fichiers,
3. démarrer le service serveur.

5.1. Installer le package

La résolution de noms est réalisée par les produits du package bind. La version actuelle est le package bind-8.x. qui remplace les versions antérieures 4.x. Dans cette version, de nombreuses modifications ont été apportées surtout au niveau de la sécurité, mais également en ce qui concerne le service DNS dynamique, c’est à dire acceptant les inscriptions des clients DHCP. La compatibilité ascendante est respectée. Nous resterons sur une configuration simple.

Si vous avez un serveur de noms déjà installé sous une version de bind 4.x, il existe une procédure de migration et de conversion des fichiers de ressources. Les outils sont cités un peu plus loin.

On va utiliser "bind 8". Il faudra donc installer les fichiers bind-8.x et bind-utils-8.x. Ce deuxième package donne quelques outils comme host, nslookup... Pour installer utilisez la commande : mount /mnt/cdrom;rpm -i /mnt/cdrom/RedHat/rpms/bind-8*.rpm

5.2. Procédure de configuration du serveur

L’installation a copié les fichiers. Sur une configuration simple vous allez avoir 5 fichiers à créer ou à modifier sur le serveur primaire :

- /etc/named.conf (fichier de configuration globale du service DNS du serveur de noms primaire),
- /var/named/db.FOO.ORG qui contiendra la description de la correspondance nom-adresse de toutes les machines du réseau
- /var/named/db.1.168.192 qui contiendra la correspondance inverse adresse-nom (pour la résolution inverse de nom in-addr.arpa), par convention db.0.168.192 si votre réseau est d’adresse 192.168.1. (db.192.168.1 est également acceptable).

- un fichier /var/named/db.LOCALHOST pour la configuration locale (localhost - 127.0.0.1).
- un fichier /var/named/db.0.0.127 pour la configuration reverse (127.0.0.1 - localhost).

Si le serveur était relié à Internet ou faisait office de serveur officiel, il y aurait d’autres fichiers à configurer.

5.3. 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 comme linuxconf. 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".

5.4. Configuration du DNS manuellement

Le fichier racine pour la configuration du serveur de noms est le fichier "/etc/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. La distribution donne un script perl qui permet de transcrire un fichier named.boot (bind version 4) au format named.conf (BIND version 8). Pour générer named.conf à partir de named.boot (si vous utilisiez une ancienne version de bind, par exemple), vous pouvez utiliser le script Perl named-bootconf qui est dans /usr/doc/bind-8.2

5.5. Le fichier named.conf

Voici un exemple de fichier commenté pour le domaine fictif archinet.edu, d’adresse 192.168.1.0.

```
#fichier named.conf pour le domaine archinet.edu
#Indication du chemin où sont localisés les fichiers de la base de données
options {
directory "/var/named";
};

#pour le fichier de cache du serveur de noms
#nous ne sommes pas raccordés sur l’Internet, donc nous ne nous servons pas de ce fichier.
# on le laisse en commentaire
# NB : trois type de commentaires dans named.conf :
# type Shell script
```

```
/* type C */
// Type C++
/*
zone "." in {
 type hint;
 file "named.ca";
};
*/

#pour la recherche directe dans le domaine, serveur primaire, on utilise le fichier
#/var/named/db.FOO.ORG
zone "foo.org" in {
 type master; # nous sommes serveur primaire de ce domaine
 file "db.FOO.ORG"; # fichier contenant les correspondances nom, adresse IP
};

# pour la recherche de zone inverse (reverse) on utilise le fichier db.1.168.192
zone "1.168.192.in-addr.arpa" in {
 type master; # nous sommes serveur primaire de ce domaine aussi
 file "db.1.168.192";
};

#pour la résolution de noms sur localhost
zone "local" in {
 type master; # nous sommes serveur primaire de ce domaine
 file "db.LOCALHOST"; # fichier contenant les correspondances nom, adresse IP
};

# rappel : la machine locale porte toujours l’adresse « localhost » 127.0.0.1
# nous proposons donc la résolution inverse sur cette zone
# la description est dans /var/named/db.0.0.127
zone "0.0.127.in-addr.arpa" in {
 type master; # nous sommes également serveur primaire de ce domaine
 file "db.0.0.127";
};
```

Notez bien que les noms appliqués aux fichiers de ressources ne sont en rien imposés. Il s’agit d’une pure convention. En effet un serveur de noms peut prendre en charge plusieurs domaines, cela permet de structurer l’organisation des fichiers de ressources.

Notez également l’option "type master". Il s’agit d’un serveur primaire. Nous verrons comment déclarer un serveur secondaire.

5.6. Le fichier /var/named/db.FOO.ORG

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. 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.

```
; NB : dans ces fichiers, les commentaires sont précédés d’un point-virgule.
; enregistrement de type SOA, on déclare tous les paramètres
; ainsi que l’adresse du responsable administratif de la zone, ici : postmaster
@  IN  SOA  ns1.foo.org. postmaster.foo.org. (
                                16 ; ces nombres ne sont pas expliqués ici
    86400 ; vous pouvez les employer tels quels
    3600 ; sans problème tant que vous ne mettez
    3600000 ; pas en place un serveur de noms
    604800 ; secondaire. )
; enregistrement de type Name Server, on déclare le serveur de noms
IN  NS  ns1.foo.org.

; on déclare les autres noeuds pour la résolution de noms
; Notez l’absence du point après les noms pour permettre « l’extension » du nom de domaine.
ns1  IN  A  192.168.1.1
; ici un client
cli1  IN  A  192.168.1.2

; on déclare les alias CNAME.  La machine proc sert de serveur de messagerie, Web, FTP, news
mail  IN  CNAME ns1
news  IN  CNAME ns1
www  IN  CNAME ns1
ftp  IN  CNAME ns1
```

5.7. Le fichier /var/named/db.LOCALHOST

```
@  IN  SOA  ns1.foo.org. postmaster.foo.org. (
                                16 ; ces nombres ne sont pas expliqués ici
    86400 ; vous pouvez les employer tels quels
    3600 ; sans problème tant que vous ne mettez
    3600000 ; pas en place un serveur de noms
    604800 ; secondaire. )

; enregistrement de type Name Server
```

```
IN  NS  ns1.foo.org.

;On déclare le noeud dans le domaine local
localhost  IN  127.0.0.1
```

5.8. Le fichier /var/named/db.1.168.192

Ici il s'agit de la résolution de noms inverse de la zone foo.org.

```
@  IN  SOA  ns1.foo.org. postmaster.foo.org. (
                                16 ; ces nombres ne sont pas expliqués ici
    86400 ; vous pouvez les employer tels quels

    3600 ; sans problème tant que vous ne mettez
    3600000 ; pas en place un serveur de noms
    604800 ; secondaire. )

; enregistrement de type Name Server
               IN  NS  ns1.foo.org.

;On déclare les noeuds dans le domaine 1.168.192.in-addr.arpa
; Ici, on ne peut pas se passer du nom complet (fini par un point).
;l'extension serait (exemple avec ns1) : 1.1.168.192.in-addr.arpa.
1  IN  PTR  ns1.foo.org.
```

5.9. Le fichier /var/named/db.0.0.127

Ce fichier assure la résolution pour 1.0.0.127.in-addr.arpa

```
@  IN  SOA  ns1.foo.org. postmaster.foo.org. (
                                            16 ; ces nombres ne sont pas expliqués ici
            86400 ; vous pouvez les employer tels quels
            3600 ; sans problème tant que vous ne mettez
            3600000 ; pas en place un serveur de noms
            604800 ; secondaire. )
)

; enregistrement de type Name Server
               IN  NS  ns1.foo.org.

;On déclare les noeuds dans le domaine 0.0.127.in-addr.arpa
```

```
; Normalement, il n’y en a qu’un : 127.0.0.1 = localhost.
; Noter le point terminal.
1  IN  PTR  localhost.
```

6. Compléments pratiques

6.1. Démarrer ou arrêter le service le service

Le service (daemon) qui active la résolution de noms s’appelle "named", prononcer "naime dé".

Si vous voulez l’arrêter ou le redémarrer dynamiquement vous pouvez utiliser les commandes suivantes :

```
/etc/rc.d/init.d/named stop
/etc/rc.d/init.d/named 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".

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 qu’il utilise lui même le service DNS et redémarrer les services réseau. Relancez, ensuite le service réseau. Utilisez les commandes suivantes :

```
/etc/rc.d/init.d/network stop
/etc/rc.d/init.d/network start
```

6.3. Procédure de configuration du client

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 Linux.

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.

6.5. Avec Linux

Vous pouvez utiliser linuxconf (cf.plus haut) ou bien 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
```

7. Procédure de tests

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, 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.

7.1. Vérifier la résolution de noms :

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/rc.d/init.d/named 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.

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.

8.1. nslookup

La commande " nslookup " est utilisée par tout administrateur ayant en charge un serveur de noms. Cette commande permet de vérifier la cohérence des enregistrements d’administration du serveur de noms. Il vaut mieux, avant de chercher à comprendre comment fonctionne une telle commande, avoir déjà cherché à installer un serveur de noms, et bien comprendre les processus mis en oeuvre.

Vous pouvez utiliser cette commande en mode interactif. Tapez nslookup puis [Entrée], vous êtes sous le prompt. Voici ce que donne l’aide en tapant le ? en guise de commande:

```
$Id: nslookup.help,v 8.4 1996/10/08 04:51:08 vixie Exp $
Commands: (identifiers are shown in uppercase, [] means optional)
NAME - print info about the host/domain NAME using default server
NAME1 NAME2 - as above, but use NAME2 as server
help or ? - print info on common commands; see nslookup(1) for details
set OPTION - set an option
all - print options, current server and host
[no]debug - print debugging information
[no]d2 - print exhaustive debugging information
[no]defname - append domain name to each query
[no]recurse - ask for recursive answer to query
[no]vc - always use a virtual circuit
domain=NAME - set default domain name to NAME
srchlist=N1[/N2/.../N6] - set domain to N1 and search list to N1,N2, etc.
root=NAME - set root server to NAME
retry=X - set number of retries to X
timeout=X - set initial time-out interval to X seconds
querytype=X - set query type, e.g., A,ANY,CNAME,HINFO,MX,PX,NS,PTR,SOA,TXT,WKS,SRV,NAPTR
port=X - set port number to send query on
type=X - synonym for querytype
class= - set query class to one of IN (Internet), CHAOS, HESIOD or ANY
server NAME - set default server to NAME, using current default server
lserver NAME - set default server to NAME, using initial server
```

```
finger [USER] - finger the optional USER at the current default host
root - set current default server to the root
ls [opt] DOMAIN [> FILE] - list addresses in DOMAIN (optional: output to FILE)
-a - list canonical names and aliases
-h - list HINFO (CPU type and operating system)
-s - list well-known services
-d - list all records
-t TYPE - list records of the given type (e.g., A,CNAME,MX, etc.)
view FILE - sort an 'ls' output file and view it with more
exit - exit the program, ^D also exits
```

Description de la commande : Nous allons voir essentiellement l'utilisation de la commande nslookup (voir les exemples commentés un peu plus bas) et de l'option "set query" La commande "set query" ou "set q", permet d'interroger un DNS, et de n'obtenir que les enregistrements que l'on désire. Par exemple la commande:

```
Set q=A renseignera sur les enregistrements de type A,
Set q= ptr renseignera sur la résolution inverse des noms dans le DNS.
```

Les outils du dns doivent être installés.

Exemple commenté d'utilisation

```
% nslookup
; voici le serveur par défaut qui va répondre.

Default Server:  ns-cache.isdnet.net
Address:  194.149.160.9
Non-authoritative answer:
isdnet.net
        origin = ns.isdnet.net
        mail addr = hostmaster.isdnet.net
        serial = 2000101301
        refresh = 28800 (8H)
        retry   = 7200 (2H)
        expire  = 604800 (1W)
        minimum ttl = 86400 (1D)
isdnet.net      nameserver = ns.isdnet.net
isdnet.net      nameserver = NS2.isdnet.net
isdnet.net      preference = 10, mail exchanger = mx-cache.isdnet.net
isdnet.net      preference = 5, mail exchanger = mailhub.isdnet.net

Authoritative answers can be found from:
isdnet.net      nameserver = ns.isdnet.net
isdnet.net      nameserver = NS2.isdnet.net
```

```
ns.isdnet.net   Internet address = 194.149.160.1
NS2.isdnet.net  Internet address = 195.154.223.1
mx-cache.isdnet.net     Internet address = 194.149.160.10
mailhub.isdnet.net      Internet address = 194.149.160.8

; je veux tous les enregistrements de la zone
> set type=ns
Server:  ns-cache.isdnet.net
Address:  194.149.160.9

; je consulte la zone wanadoo.fr
> wanadoo.fr
Server:  ns-cache.isdnet.net
Address:  194.149.160.9

Non-authoritative answer:
wanadoo.fr      nameserver = ns2.wanadoo.fr
wanadoo.fr      nameserver = ns2.wanadoo.com
wanadoo.fr      nameserver = ns.wanadoo.fr
wanadoo.fr      nameserver = ns.wanadoo.com

Authoritative answers can be found from:
ns2.wanadoo.fr  Internet address = 193.252.19.11
ns2.wanadoo.com Internet address = 194.51.238.2
ns.wanadoo.fr   Internet address = 193.252.19.10
ns.wanadoo.com  Internet address = 194.51.238.1

; je vais solliciter le serveur de noms ns2 de wanadoo.fr
> server ns2.wanadoo.fr
Default Server:  ns2.wanadoo.fr
Address:  193.252.19.11

; je veux les enregistrements de type MX
> set type=mx
> wanadoo.fr
Server:  ns2.wanadoo.fr
Address:  193.252.19.11

wanadoo.fr      preference = 20, mail exchanger = dixon.rain.fr
wanadoo.fr      preference = 10, mail exchanger = smtp.wanadoo.fr
wanadoo.fr      nameserver = ns.wanadoo.fr
wanadoo.fr      nameserver = ns.wanadoo.com
wanadoo.fr      nameserver = ns2.wanadoo.fr
wanadoo.fr      nameserver = ns2.wanadoo.com
dixon.rain.fr   Internet address = 194.51.3.36
```

```
smtp.wanadoo.fr Internet address = 193.252.19.163
ns.wanadoo.fr   Internet address = 193.252.19.10
ns.wanadoo.com  Internet address = 194.51.238.1
ns2.wanadoo.fr  Internet address = 193.252.19.11
ns2.wanadoo.com Internet address = 194.51.238.2

; Je vais évaluer la résolution de noms inverse
; en demandant des enregistrement de type ptr
> set type=PTR
; quel le nom de la machine qui a l'adresse 10.19.252.193
> 10.19.252.193.in-addr.arpa
Server:  ns2.wanadoo.fr
Address:  193.252.19.11

10.19.252.193.in-addr.arpa      name = ns.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns.wanadoo.com
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.fr
19.252.193.in-addr.arpa nameserver = ns2.wanadoo.com
ns.wanadoo.fr   Internet address = 193.252.19.10
ns.wanadoo.com  Internet address = 194.51.238.1
ns2.wanadoo.fr  Internet address = 193.252.19.11
ns2.wanadoo.com Internet address = 194.51.238.2

; on va voir si la résolution de noms d'un autre domaine fonctionne
; peux tu me dire également quel est le nom de la machine d'adresse 4.90.115.195

> 4.90.115.195.in-addr.arpa
Server:  ns2.wanadoo.fr
Address:  193.252.19.11

4.90.115.195.in-addr.arpa       name = argo.beaupeyrat.com
90.115.195.in-addr.arpa nameserver = argo.beaupeyrat.com
90.115.195.in-addr.arpa nameserver = pw.beaupeyrat.com
argo.beaupeyrat.com     Internet address = 195.115.90.4
pw.beaupeyrat.com       Internet address = 195.115.90.3
^D
```

Ouf, pour cette fois tout va bien.

Il semble que la commande "nslookup" est de plus en plus abandonnée au profit de la commande "dig".

8.2. 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 -HINT ’PID de named’". Vous récupérez un fichier /var/named/named_dump.db que vous pouvez exploiter.

8.3. 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 /etc/rc.d/init.d/named stop
- Nettoyage du fichier cat /dev/null > /var/log/messages
- Démarrage du serveur
- Consultation des logs : cat /var/log/messages | more

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.

10. Annexes

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 1 enregistrement de type pointeurs (PTR) dans la zone de reverse : in-addr.arpa. Enfin, pour le serveur de messagerie, il faut également un enregistrement de type MX.

```
; Exemple de ZONE DIRECTE
;      Zone version:  98030104l
;

@                       IN SOA ns1.foo.org. postmaster.mail.foo.org. (
                        98030104    ; serial number
                        21600       ; refresh
                        3600        ; retry
                        604800      ; expire
                        3600      ) ; minimum TTL

;
;  Zone NS records
;

@                       IN NS ns1
```

```
;
;  Zone records
;

@                       IN MX 10 mail
ns1   IN A 192.168.0.1
mail                    IN CNAME mailhost
mailhost                IN A 192.168.0.2
www                     IN CNAME ns1
ftp                     IN CNAME ns1
;************************************************************
; Exemple de ZONE REVERSE
;
;      Zone version:  4l
;

@                       IN SOA ns1.foo.org. postmaster.mail.foo.org. (
                        4           ; serial number
                        3600        ; refresh
                        600         ; retry
                        86400       ; expire
                        3600      ) ; minimum TTL
;
;  Zone NS records
;

@                       IN NS ns1.foo.org.
;
;  Zone records

1                       IN PTR ns1.foo.org.
2                       IN PTR mailhost.foo.org.
```

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-dessous. 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". Vous devrez donc avoir dans /var/named les fichiers "db.LOCALHOST" et "db.0.0.127".

```
options {
 directory "/var/named";
```

```
};

zone "foo.org" in {
 type slave;
 file "db.foo.org";
 masters { 192.168.0.1; } ;
/* ici on met l’adresse du primaire */

};
/* pour la zone reverse */
zone "0.168.192.in-addr.arpa" in {
 type slave;
 file "db.192.168.0";
 masters { 192.168.0.1; } ;
}

/*pour la résolution de noms sur localhost*/
zone "local" in {
 type master; # nous sommes serveur primaire de ce domaine
 file "db.LOCALHOST"; # fichier contenant les correspondances nom, adresse IP
};

/* le secondaire pour la zone foo.org est primaire
  pour la zone locale */
zone "0.0.127.in-addr.arpa" in {
 type master;
 file "db.0.0.127";
};
```

Vous avez remplacé la définition "masters" par "slave" sauf pour db.LOCALHOST et db.0.0.127 qui sont lus localement. Ensuite vous avez rajouté l’adresse du serveur à partir duquel le transfert de zone doit s’effectuer. Les noms des fichiers servent au serveur de noms secondaire pour réaliser ses sauvegardes.

Activer le serveur de noms esclave et analyser les log. 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.

*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. Relancez ensuite le service sur le serveur de noms ns2. 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 1/2 heure. 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 1/2 h après la synchronisation s’opérer. 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 dans /var/named, 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.

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 gauche il faut rajouter les enregistrements qui décrivent les serveurs de noms de la zone sd.foo.org dans le fichier 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 gauche 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.
```

Installation d’un serveur DNS - TP

La résolution de noms - Fiche de TP

Alix MASCRET

Mode d’utilisation du serveur DNS Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Installation du service DNS

1.1. 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 nslookup

4. Quatrième partie : mise en place du serveur secondaire, modification de l’enregistrement SOA du serveur primaire. :

   Test du transfert de zone

1.2. Le contexte

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.

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

Pour la configuration des machines et les procédures de test, vous utiliserez les documents fournis dans la fiche de cours et en annexe.

1. Annexe 1 : fichier /etc/named.conf pour un dns primaire
2. Annexe 2 : fichier de zone /var/named

3. Annexe 3 : fichier /etc/named.conf pour un dns secondaire
4. Annexe 4 : fichier /etc/resolv.conf
5. Annexe 5 : utilisation de nslookup

1.3. 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.

Installez le package

On va utiliser le "bind 8". Il faudra donc installer les fichiers *bind-8.x* et *bind-utils-8.x*. Ce deuxième package donne quelques outils comme nslookup.

1.4. Installation du serveur de noms primaire

Procédure de configuration du serveur

l’installation a copié les binaires et les scripts de lancement du daemon « named » dans /etc/rc.d/init.d. Elle a également créé un compte utilisateur « named » pour faire tourner le daemon sous ce compte et non sous le compte « root ». Attention, si vous créez des fichiers ou répertoire sous le compte « root », le processus doit pouvoir lire les fichiers et écrire dans les répertoires.

Vous allez avoir 5 fichiers à créer sur le serveur primaire :

- -/etc/named.conf (fichier de configuration du serveur de noms primaire),
- -/var/named/db.COULEUR.ORG qui contiendra la description de la correspondance nom-adresse de toutes les machines de votre zone.
- - /var/db.192.168.x.0 qui contiendra la correspondance inverse adresse-nom (pour la résolution inverse de nom in-addr.arpa), par exemple db.192.168.x si votre réseau est d’adresse 192.168.x.
- - un fichier /var/named/db.LOCALHOST pour la configuration locale (localhost - 127.0.0.1).
- - un fichier /var/named/db.127.0.0 pour la configuration reverse (127.0.0.1 - localhost).

Configuration du service serveur DNS manuellement

Faites une copie de sauvegarde des fichiers proposés par défaut si la procédure d'installation en a installé. /etc/named.conf et tous les fichiers de /var/named.

Créez le fichier /etc/named.conf pour votre zone à partir de celui fourni en annexe. Ne prévoyez dans un premier temps, que la zone directe de votre domaine, la zone directe de « localhost ».

Créez les fichiers de ressources pour la zone dans /etc/named.conf

1.4.1. Démarrer ou arrêter le service le service

Si vous voulez l'arrêter ou le redémarrer dynamiquement vous pouvez utiliser les commandes suivantes:

/etc/rc.d/init.d/named stop | stop ...

ou mieux kill -HUP PID_DE_NAMED

Avant de continuer vérifiez :

que le service est bien démarré « ps aux | grep named »

qu'il n'y a pas d'erreurs dans les journaux (fichiers .log) « cat /var/log/messages | less »

Corrigez et n'allez pas plus loin tant que tous les problèmes ne sont pas résolus. Si le fichier d'erreurs est trop important, vous pouvez le réinitialiser « cat /dev/null > /var/log/messages ».

1.4.2. Configuration 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 » en adaptant celui fourni en annexe.
- Relancez le service réseau « /etc/rc.d/init.d/network restart »
- 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.

1.5. Configuration de la zone reverse

Configurez à ’aide des fichiers fournis en annexe la zone inverse (reverse). Ceci consiste à rajouter une déclaration dans le fichier « /etc/named.conf », une pour la zone locale « localhost » et une pour la zone « couleur.org » et à créer les deux fichiers correspondant dans « /var/named »

Relancez le service « named », vérifiez les journaux, corrigez les éventuelles erreurs.

Vérifiez à ’aide de « nslookup » que la résolution de noms inverse fonctionne.

1.6. Installation du serveur de noms secondaire

Sur M2 créez le fichier « /etc/named.conf ». Créez également dans « /var/named » les fichiers pour la zone locale « localhost » car tout serveur, même secondaire est primaire pour « sa » zone locale.

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/messages », une trace qui confirme le transfert de zone.

N’allez pas plus loin tant que cela n’est pas en parfait état de fonctionnement.

1.6.1. Procédure de test du serveur secondaire

Arrêtez sur le serveur primaire le service named « /etc/rc.d/init.d/named stop »

Testez le fonctionnement du serveur secondaire 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.

1.7. Test de l’enregistrement SOA

Modifiez au minimum le temps de rafraîchissement des enregistrements du serveur Primaire. Modifiez également le N° de série. Relancez le serveur secondaire et vérifiez dans les logs que le transfert de zone s’effectue bien.

2. Annexes et documentation complémentaire

Voir fiche de cours.

Installation d’un serveur NFS

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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 installer un service serveur NFS, et comment utiliser le service client.

2. Installation des produits clients et serveurs

Les services NFS sont activés par les daemons rpc.mountd et rpc.nfsd. Ils sont exécutés au démarrage du système. Sinon ils peuvent être activés par la commande : /etc/rc.d/init.d/nfs start.

Ces deux programmes s’appuient sur des 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 ns1

```
program  vers  proto    port
100000     2    tcp     111   rpcbind
100005     2    tcp     635   mountd
100003     2    tcp    2049   nfs
```

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 -rw cli

#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. Configuration du serveur

Vérifiez que le noyau supporte le système de fichiers nfs:

Utilisez la commande more /proc/filesystems, voici ce que vous pouvez obtenir.

ext2

msdos

nodev proc

nodev nfs

Le système de fichiers nfs doit apparaître.

2.3.1. Le fichier /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 fichiers

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 fichier /etc/mtab - exemple de fichier avec la commande "more /etc/mtab"

Ce fichier 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 : 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

Installez les packages nfs et portmap

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. A 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 serveur de messagerie

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Résumé

Le document décrit l’installation des services SMTP et POP3 afin de mettre en place un serveur de messagerie. Nous verrons également comment mettre un service de liste de messagerie de base avant d’avoir un service plus spécialisé.

Il existe de nombreux produits sous licence libre comme sendmail, postfix ou qmail par exemple. Toutes les distributions de Linux disposent d’au moins un serveur de messagerie. Chacun présente des avantages, des inconvénients et chacun a ses défenseurs et ses détracteurs. Je ne parlerai pas de cela ici.

Nous allons voir le principe d’installation d’un serveur de messagerie et d’un client de messagerie. Les principes seront décrits avec les produits sendmail et postfix.

2. Installation de Sendmail

L’installation concerne sendmail, plus précisément la version de Jussieu que vous pouvez télécharger sur ftp.jussieu.fr.

Pourquoi sendmail ?

Si vous êtes amené à administrer des serveurs sous Unix/Linux, vous serez un jour confronté à sendmail.

Pourquoi le kit de Jussieu ?

Il y a de nombreuses raisons, mais parmi les principales je citerai son support (notamment pour la sécurité) et son excellente documentation.

Les services de messagerie sont parmi les services réseaux les plus complexes à configurer pour de nombreuses raisons :

- utilisation de réseaux étendus et souvent très hétérogènes,
- nécessite des droits privilégiés pour l'authentification d'où des risques possibles de failles,
- besoin de filtrer les abus de messagerie (spams).

Dans les niveaux de complexité des logiciels de messagerie, Sendmail, tient la première place. Le kit de Jussieu propose un "configurateur" qui simplifie un peu la procédure de configuration, mais surtout propose une méthodologie pour l'organisation des domaines et des serveurs.

Vous pouvez bien sûr, si vous utilisez des outils comme "linuxconf" préférer le produit standard de la distribution. Pour ma part je considère toujours que la configuration d'un service en environnement graphique est toujours plus simple une fois qu'il a été configuré au moins une fois à la main, mais cet avis est personnel.

A la fin de ce document :

- Vous aurez un serveur de messagerie complètement opérationnel pour un intranet,
- Vous pourrez mettre en place un service de liste de diffusion élémentaire,
- Vous ne maîtriserez pas complètement ce service complexe, mais vous aurez mis un pied dans ce que les administrateurs de réseau prennent pour un des domaines les plus complexes de leur métier.

2.1. MHS, MTA, UA

Le MHS, Message Handler System est le système global de messagerie,

Le MTA, Message Transfert Agent est composé de deux agents. Un agent de routage et un agent de transport.

L'agent de routage a pour but d'acheminer le message, en fonction de l'adresse de 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, le protocole de transport peut être SMTP ou UUCP. Sendmail assure les deux fonctions de transport et de routage.

L’UA, user Agent, ou MUA Message User Agent, est le programme utilisé par le client pour composer, envoyer et recevoir les messages. Il existe également un agent (UA) pour la remise physique du courrier entrant dans la boîte aux lettres de l’utilisateur. Sur Linux nous utilisons "procmail". Pour la composition 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". Les protocoles utilisés sont SMTP ou UUCP pour envoyer, et POP3 ou IMAP pour recevoir. Il y a au moins un UA à chaque MTA.

2.2. Installation

Téléchargez la dernière version du kit sur ftp.jussieu.fr. Vous devez récupérer les documents suivants :

- le binaire compilé ou les sources de la dernière version de sendmail pour votre système d’exploitation,
- le configurateur et ses exemples,
- la documentation.

Décompressez les fichiers. Vous allez avoir dans la liste les fichiers suivants :

- Sendmail, binaire servant de serveur SMTP,
- Sendmail.hf, fichier d’aide de sendmail,
- Sendmail.st, fichier de statistiques,
- Mailstat, programme affichant les statistiques.

Les répertoires :

/var/spool/mqueue, file d’attente d’envoi des messages,

/var/spool/mail, boîtes aux lettres des utilisateurs.

Localisez ces fichiers sur votre serveur :

1. Notez bien les droits, propriétaires et groupes de chaque fichier avec la commande ls-al,
2. Faites une sauvegarde de chacun de ces fichiers avec "mv",
3. sauvegardez également le fichier /etc/sendmail.cf.

Voici un extrait de ce que vous devriez trouver :

```
-rw-r--r--    1 root     root            0 jui  8  1999 /var/log/sendmail.st
lrwxrwxrwx    1 root     root           16 mai 22 15:43 /usr/lib/sendmail -> ../sbin/sendmai
-r--r--r--    1 root     root         5342 jui  8  1999 /usr/lib/sendmail.hf
-rwsr-sr-x    1 root     root       324380 jui  8  1999 /usr/sbin/sendmail
```

2.3. Fonctionnement du configurateur

Le "configurateur" est un script shell qui va générer le fichier de configuration de sendmail "sendmail.cf". La génération de sendmail.cf va tenir compte d’un certain nombre de paramètres :

ceux qui définissent la configuration de votre site (nom de domaine, nom du serveur de messagerie.)

ceux qui définissent les règles que vous désirez appliquer lors de l’envoi ou la réception de message (par exemple des règles de réécriture des adresses : user@NomMachine.NomDomaine en user@NomDomaine.)

Pour décrire ces paramètres vous avez le choix entre deux solutions :

la première consiste à décrire les paramètres dans deux fichiers textes séparés, un pour la configuration, un pour les règles. On utilisera le configurateur sous la forme "./configurateur FichierDeRègles FichierDeConfiguration > sendmail.cf",

la deuxième consiste à décrire la configuration dans un fichier texte séparé, et les règles à l’intérieur même du configurateur. L’utilisation sera alors de la forme : "./configurateur FichierDeConfig > sendmail.cf".

Nous allons utiliser la première option.

Vous devrez pour cela modifier le script "configurateur" et créer les deux fichiers textes. Les parties modifiées du script sont données complètement en annexe. Les paragraphes modifiés sont indiqués. Les fichiers de règles et de configuration sont décrits dans la section suivante.

2.4. Création du fichier de "config" et du fichier de "règles"

```
#------------------------ Fichier de config ---------------------------
#------------------------ archinet.config ---------------------------
# Nom ou alias de la machine déclaré dans le DNS
Host=’mail’
# Stocker les messages ?
Mailhost=’o’
#Localisation de la base de données des alias
Aliases=’/etc/aliases’
```

```
#Localisation du fichier de stat
SendmailSt=’/var/log/sendmail.st’
#Localisation du fichier d’aide
SendmailHf=’/usr/lib/sendmail.hf’
#Localisation de la file d’attente
Mqueue=’/var/spool/mqueue’
#Utilise ou non la version 8 de sendmail
V8=’o’
# Arguments à utiliser pour le mailer local. Sous Linux on utilise procmail
MailerLocal=’/usr/bin/procmail lsDFMShPfn procmail -a $h -d $u’


#---------------------- Fichier de règles --------------------------
#------------------------ archinet.regles --------------------------
#!/bin/bash
#Nom du domaine
SITE=archinet.edu
Domaine=$SITE
#Aucun relais extérieur
RelaisExterieur=
#Tout ce qui est à destination de *.$DOMAINE ou $DOMAINE reste sur cette machine
AdressesLocales=TOUT_DOMAINE
#Doit on redistribuer du courrier vers d’autres domaines, non.
AdressesInternes=RIEN
#les adresses user@$HOST.$DOMAINE sont réécrites en user@$DOMAINE
ReecritureAdressesLocales=archinet.edu
```

2.5. Création du fichier sendmail.cf

Il ne reste plus qu’à générer le fichier sendmail.cf. Pour cela vérifiez que le script est en mode exécutable et utilisez la commande :

"./configurateur archinet.regles archinet.config > sendmail.cf".

Installation des fichiers

Puisque tout est prêt vous allez installer les fichiers. Sous Linux cela doit donner :

1. Copiez le sendmail dans /usr/sbin

   Vous allez également vérifier les droits de ces fichiers.

   Modifiez les droits "chmod 6755 sendmail", cela doit donner :

-rwsr-sr-x 1 root root 400460 jun 18 15:22 sendmail

2. Copiez sendmail.hf dans /usr/lib
3. Copiez mailstat dans /usr/sbin
4. Copiez sendmail.cf dans /etc ou dans /etc/mail en fonction de la version compilée
5. Activez sendmail avec la commande :
6. "/etc/rc.d/init.d/sendmail start | stop | status | restart"

Vous devrez également configurer le daemon sendmail pour qu’il soit activé à chaque démarrage.

Création des alias

L’utilisation des alias est fréquente par les administrateurs à plusieurs titres. Ils permettent par exemple d’associer l’alias jean.dupont au compte d’utilisateur système créé "dup007". Il sera ainsi possible d’écrire à jean.dupont@archinet.edu plutôt que dup007@archinet.edu.

Ce procédé est également utilisé pour les comptes systèmes root, postmaster, webmaster, abuse, hostmaster. Ils permettent de rediriger les messages adressés à abuse@archinet.edu vars jean.dupont@archinet.edu qui lui même relève les messages de la boîte dup007@archinet.edu.

Comment créer un alias ? Ce n’est pas compliqué.

- Créez un compte utilisateur système.
- Ouvrez à l’aide d’un éditeur le fichier /etc/aliases, vous aurez déjà des indications,
- Créez un ou plusieurs alias pour ce compte système,
- Utilisez la commande "newaliases" pour mettre à jour la base de données indexée des alias "/etc/aliases.db".

Si la procédure vous semble fastidieuse, vous pouvez utiliser un script qui crée l’alias automatiquement lors de la création d’un compte utilisateur car il ne s’agit que de fichiers textes.

Création d’une liste

Attention, ici il ne s’agit pas d’un gestionnaire de liste comme majordomo, sympa ou autres. Les listes s’appuient sur le même principe, à la différence près qu’il ne s’agit pas d’une personne mais d’un groupe de personne.

Pour créer une liste il faut déclarer 2 éléments dans le fichier /etc/aliases:

1. la liste avec ses membres. Chaque membre est séparé par une virgule.
2. Le responsable qui aura la gestion de cette liste. Il recevra les messages d'erreurs, si un problème de résolution de noms est rencontré.

Voici la structure des enregistrements:

```
Owner-NomDeLaListe: NomDuCompte
NomDeLaListe:  User1,... Usern

Pour nous cela va donner:
owner-UneListe:  AdmUneListe
UneListe:  U1,U2,U3
```

Le fichier est terminé, il ne reste plus qu'à le compiler. Ceci dit, le problème d'administration saute aux yeux. La maintenance risque d'être très vite invivable, si le site compte plusieurs centaines de personnes. Pour éviter cela, on peut utiliser la commande "include NomDeFichier" et mettre les noms à gérer dans des fichiers à part. Vous pouvez créer un fichier texte par liste dans /var/mail/log qui contient la liste des personnes appartenant à cette liste. Voici un exemple de déclaration :

```
/var/mail/log/ListeA
/var/mail/log/ListeB
```

Chacune comprenant la liste des comptes appartenant à la liste. Par exemple si on réalise un "cat ListeA" on peut avoir :

```
U1
U2
U3
etc, etc.
```

Notre gestion de liste deviendrait :

```
owner-All:  AdmAll
ListeAll:  :include/var/mail/log/ListeA, include/var/mail/log/ListeB

owner-ListeA:  AdmListeA
ListeA:   :include/var/mail/log/ListeA

owner-ListeB:  AdmListeB
ListeB:   :include/var/mail/log/ListeB
```

Commentaires :

Pour chaque message envoyé à ListeAll@archinet.edu les utilisateurs recevront une copie.

AdmAll, AdmListeA, AmdListeB sont des alias. Ils peuvent représenter la même personne. Cela simplifie l’administration en cas de départ ou mouvement de personnel.

Ce mode d’administration des listes devient beaucoup plus souple.

Pour compiler le nouveau fichier, tapez la commande "newaliases"

Le programme vous donne le résultat pour les ajouts. Vous pouvez également consulter l’aide avec la commande "man aliases"

Même remarque que pour la création des alias individuels, vous pouvez adapter le script de création de compte d’utilisateur pour affecter automatiquement le compte créé à une ou plusieurs listes.

2.6. Modification du serveur de noms

Nous allons utiliser le serveur de noms ns1.archinet.edu comme serveur de messagerie. Nous avons vu que les règles de réécriture d’adresse vont procéder à deux remplacements:

remplacement de l’alias du compte utilisateur par son compte,

réécriture de archinet.edu par NomDuServeurDeMessagerie.archinet.edu.

Nous voulons que les adresses soient adressées à user@mail.archinet.edu et non user@ns1.archinet.edu. Pour cela il faut créer dans la base des enregistrements du serveur de noms un enregistrement de type CNAME :

mail IN CNAME ns1.archinet.edu,

Mais ce n’est pas tout, il faut également signifier que ns1, ou mail est également relais de messagerie. Nous allons donc créer un enregistrement de ressources de type MX (Mail eXchanger):

ns1 IN MX 10 192.168.1.1

Que signifie le "10" ?

Il est à rapprocher du principe de "redondance" appliqué sur Internet afin de garantir un service permanent. Tout comme vous devriez avoir normalement un DNS primaire et un DNS secondaire, vous devriez avoir un serveur de messagerie prioritaire et un serveur de messagerie secondaire. Dans ce cas il y aurait deux enregistrements, chacun ayant un niveau de priorité. Par exemple :

SrvMsg1 IN MX 10 192.168.1.1

SrvMsg2 IN MX 50 192.168.1.2

Relancez le service de résolution de noms.

2.7. Le SPAM

Le "SPAM" est un procédé ou plus précisément une calamité qui consiste à utiliser les services de messagerie comme vecteur de diffusion de la publicité. Dans le cas d’un intranet, une protection de ce style n’a pas d’intérêt. Ceci dit le mal existe, il faut s’en protéger, voire le combattre. Les administrateurs vous diront que ces actes peuvent mettre à mal un serveur de messagerie, c’est à dire tout un système, nous y sommes donc fortement sensibilisés. Vous trouverez une excellente documentation dans le Kit de Jussieu et également sur www.nic.fr. Vous trouverez également les procédures à mettre en oeuvre si vous êtes victime de ce type d’actions.

2.8. Configuration du serveur 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".

/etc/inetd.conf

Pop and imap mail services et al

#

#pop-2 stream tcp nowait root /usr/sbin/tcpd ipop2d

pop-3 stream tcp nowait root /usr/sbin/tcpd ipop3d

#imap stream tcp nowait root /usr/sbin/tcpd imapd

#

Vous pourrez également sécuriser le service avec TCPWrapper si vous l’utilisez. Relancez inetd.

Attention, sur certains environnements, vous devrez également mettre les permissions "rwx" sur "/tmp" pour tout le monde. Si vous n’arrivez pas à relever vos messages, le problème peut venir de là.

2.9. Configuration d’un client de messagerie

Les clients de messagerie (Eudora, Netscape, Outlook...) vont tous utiliser les mêmes paramètres. Leur configuration est également assez simple.

Serveur SMTP (envoi de courrier) mail.archinet.edu, (port 25)

Serveur Pop3 (réception de courrier) mail.archinet.edu. (port 110)

Le serveur pop authentifiera les connexions des utilisateurs pour relever les messages.

2.10. Procédure de test

Créez un compte utilisateur pn,

Créez un alias prenom.nom pour ce compte système dans le fichier /etc/aliases.

Les messages envoyés à :

pn@ns1.archinet.edu, pn@mail@archinet.edu, pn@archinet.edu

prenom.nom@ns1.archinet.edu, prenom.nom@mail.archinet.edu, prenom.nom@archinet.edu

doivent tous être correctement délivrés.

2.11. Procédure de débogage

Vous avez une procédure qui permet de tester les règles dans la documentation du kit de Jussieu. Vous pouvez toutefois, si vous rencontrez des problèmes vérifier que :

le service DNS est actif et en parfait état de fonctionnement, c’est à dire que la résolution de noms est opérationnelle, notamment pour le nom "mail.archinet.edu,

le service pop3 dans inetd.conf est activé,

les services ont bien été relancés après la modification des fichiers de configuration,

la commande newaliases a été utilisée après la création des alias.

Vérifiez également les fichiers /var/log/messages, /var/maillog. Vous pouvez avoir des indications sur les dysfonctionnements éventuels.

Si vous avez un accès sur l’extérieur, l’université de Rennes met à la disposition des administrateurs un robot qui permet de tester le fonctionnement de la messagerie. Envoyez un mail à : echo@univ-rennes1.fr

2.12. Conclusion

Il n’y a plus grand chose à voir sur une configuration de base, par contre cela se complexifie très vite dès que l’on désire segmenter le réseau en département (plusieurs serveurs de messagerie), sécuriser un serveur de messagerie public à l’aide d’un firewall. Cet exercice de mise en exploitation d’un service de messagerie, présente toutefois l’avantage d’approcher d’assez près la problématique de l’échange de données par messagerie électronique.

2.13. Annexe : Configurateur pour le domaine archinet.edu

Vous avez ici, la liste des modifications à apporter au configurateur pour le domaine archinet.edu. Il y a peu de choses à faire. Tout est dans le début du script.

```
#!/bin/sh

# set -x

#
# Configurateur de sendmail.cf pour les laboratoires de jussieu.fr
# ou d’ailleurs.
#
# Auteurs :
#   Jacky Thibault (jt@ccr.jussieu.fr)
#   Pierre David (pda@prism.uvsq.fr)
#
# Historique
#   91/04/03 : pda/jt : conception
#   91/11/28 : pda/jt : version 1.2   : migration de la liaison bitnet du CCR
#   92/01/29 : pda/jt : version 1.3   : adaptation a sendmail 5.65 et a IBM
#   92/07/17 : pda/jt : version 1.4   : simplification
#   93/01/26 : pda/jt : version 1.4   : remplacement de D par M pour IBM
#   93/02/02 : pda/jt : version 1.5   : renommage des regles 8, 9 et 29
#   93/06/09 : pda    : version 1.6   : parametrage du site (pas jussieu en dur)
#   93/11/26 : pda    : version 1.7   : adaptation a sendmail V8
#   93/11/26 : pda    : version 1.7   : fusion des sendmail.cf
#   93/11/26 : pda    : version 1.7   : ajout des revaliases
#   94/05/03 : pda/jt : version 2.0   : generalisation
#   94/06/23 : pda/jt : version 2.1   : corrections
#   94/12/08 : pda/jt : version 2.1   : correction du bug dans la regle 16
#   94/12/08 : pda/jt : version 2.1   : retrait de canonisation superflue ds S19
#   94/12/19 : pda/jt : version 2.1   : retrait de la variable dbm
#   94/12/19 : pda/jt : version 5.1   : version homogene a celle du sendmail.cf
#   95/02/07 : pda    : version 5.1.1 : correction d’un pb dans les ad. int.
#   95/03/08 : pda    : version 5.1.2 : correction signalee par R. Negaret
#   95/06/28 : pda/jt : version 5.1.3 : revaliases multi-domaines
#   95/11/07 : pda/jt : version 5.2   : conversion pour sendmail 8.7
#   95/11/07 : pda/jt : version 5.2   : disparition des adresses *.uucp
#   96/01/18 : pda/jt : version 5.2   : changement du relais bitnet
#   96/01/18 : pda/jt : version 5.2   : recherche de cpp
#   96/02/01 : pda    : version 5.2   : ajout de SendmailHf et Mqueue
#   96/02/08 : pda/jt : version 5.2   : ajout de MailerLocal
#
```

```
# Certains systemes ne mettent pas cpp a l'endroit habituel (/lib/cpp)
# /lib          : toutes les machines classiques
# /usr/ccs/lbin : HP-UX 10.x
# /usr/ccs/lib  : Solaris 2.x
# /usr/libexec  : BSD 4.4 (FreeBSD)
# /usr/bin      : Linux

LCPP="/bin /usr/bin /lib /usr/lib /usr/ccs/lbin /usr/ccs/lib /usr/libexec"
CPP=""
for i in $LCPP
do
    if [ -f $i/cpp ]
    then
#---------------------------- Modifié-----------------------------
 CPP=$i/cpp
 # Pour Linux et NeXT, mettre la ligne suivante :
 CPP="$CPP -traditional"
#-----------------------------------------------------------------
 break
    fi
done

if [ -z "$CPP" ]
then
    echo "Le preprocesseur du langage C (cpp) est utilise," >&2
    echo "mais il n'est pas trouve sur votre machine." >&2
    echo "Editez le fichier 'configurateur' et modifiez la variable CPP" >&2
    echo "qui se trouve en debut." >&2
    exit 1
fi

##########################################################################
# Premiere option :
# - utiliser un fichier de regles du site separe et l'appeler ici.
##########################################################################



#---------------------------- Modifié-----------------------------------
# Décommentez les lignes de la première option car on utilise deux fichiers
#---------------------------- Modifié-----------------------------------

# #
# # Syntaxe : configurateur fichier-regles fichier-configuration
```

```
# #
#
 if [ $# != 2 -o ! -f "$1" -o ! -f "$2" ]
 then
     echo "usage : $0 fichier-regles fichier-configuration"  >&2
     exit 1
 fi
#
# #
# # Les scripts sont executes dans le contexte du shell courant et pas dans
# # un sous-shell. Ceci necessite l'execution par la commande "." (point).
# #
#
# #
# # Executer le fichier de configuration
# # (donner les variables de la station)
# #
 chmod +x $2
 . $2

# #
# # Executer le fichier de regles du site
# # (traduire les variables de la station en fonction des variables
# # de bas niveau du configurateur)
# #
 chmod +x $1
 . $1


#########################################################################
# fin de la première option
#########################################################################

#########################################################################
# Deuxieme option :
# - integrer un fichier de regles du site a cet endroit et donner ainsi
#   un package simple d'emploi.
# - dans ce cas :
# commentez la premiere option ci-dessus (pas d'appel de fichier separe),
# decommentez les lignes ci-dessous
# inserez vos regles a l'endroit indique ci-dessous
#########################################################################


#---------------------------- Modifié-------------------------------------
```

```
# Commentez les lignes de la deuxième option car on utilise deux fichiers
# Ici j’ai tout supprimé
#---------------------------Modifié-------------------------------------


########################################################################
# fin de la deuxieme option
########################################################################
```

Dans le reste, rien n’a été modifié.

3. Installation de Postfix

Postfix ne s’appuie pas sur la configuration d’un programme unique. Pour chaque fonction il propose un programme.

- Lecture des messages locaux
- Réception SMTP
- Réécriture d’adresse
- Envoi SMTP
- Traitement des erreurs
- ...

Cela améliore grandement la sécurité et la supervision des actions des programmes.

Les systèmes gèrent également plusieurs files d’attentes (maildrop - messages locaux, deferred - messages en attente, active - messages en cours ou en attente de transport...)

3.1. Installation de Postfix à partir des sources

Postfix est un programme encore qualifiable de récent (juin 2001). Cela signifie qu’il vaut mieux télécharger la dernière version sur le site officiel et réaliser l’installation à partir des sources. La procédure ne pose pas de difficulté majeure. Sinon vous pouvez toujours l’installer à partir de packages

pré-configurés. Pour le TP vous utiliserez les packages Linux pré-configurés qui évitent quelques manipulations.

Les commandes sont dans bin/, les daemons dans libexec/ , les fichiers de configuration dans etc/, les files d'attentes dans var/.

3.2. Les fichiers de configuration

master.cf

```
              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
      pickup    fifo  n       n       y       60      1       pickup
      cleanup   unix  -       -       y       -       0       cleanup
      qmgr      fifo  n       -       y       300     1       qmgr
      rewrite   unix  -       -       y       -       -       trivial-rewrite
      bounce    unix  -       -       y       -       0       bounce
      defer     unix  -       -       y       -       0       bounce
      smtp      unix  -       -       y       -       -       smtp
      showq     unix  n       -       y       -       -       showq
      local     unix  -       n       n       -       -       local
      cyrus     unix  -       n       n       -       -       pipe
      flags=R user=cyrus argv=/usr/cyrus/bin/deliver -e -q -m ${extension} ${user}
      #uucp     unix  -       n       n       -       -       pipe
          flags=F user=uucp argv=uux -n -z -a$sender - $nexthop!rmail ($recipient)
```

main.cf

```
              Définit toute la configuration
              parametre = valeur
              paramètres multilignes
              version francisée

         Utilitaire postconf
              -d : donne les paramètres par défaut
              -n : donne les paramètres changés
```

Exemple de configuration de master.cf

Voici une liste de variables à configurer dans le master.cf, afin de mettre en place un service minimum.

```
mydestination=host.domaine.fr
myorigin=domaine.fr
masquerade_domains=domaine.fr
relayhost=smtp:[mailhost.domaine.fr]
disable_dns_lookups=yes
```

3.3. Les files d'attentes

Lorsqu'on a l'habitude d'utiliser sendmail, la multiplicité des files d'attente de postfix a tendance à dérouter. En effet, en lieu et place du classique et unique /var/spool/mqueue/ on se retrouve avec l'arborescence suivante :

```
# tree /var/spool/postfix/
/var/spool/postfix/
|-- active
...

|-- deferred
|   |-- 048ED779D1
|   |-- 09085779E0
|   |-- 17101779D6
|   |-- 31DE0779DA
|   |-- 3D15D779D4
|   |-- 4D2BD779D7
|   |-- 7BFA6779D3
|   |-- 7C65D779CF
|   |-- B10C3779D0
|   `-- C3272779D2
...

|-- incoming
...

|-- maildrop
...
```

Nous avons abrégé la sortie de cette commande pour ne retenir que les noms de répertoires qui nous intéressent ici. postfix utilise quatre files d'attentes :

1. maildrop contient les messages locaux ;

2. incoming contient les messages qui ont été prélevés dans maildrop par le démon pickup, puis qui ont été traités par le démon cleanup. Cette file contient aussi les messages venant de l'extérieur. En bref, elle contient les messages qui n'ont pas encore été traités par le gestionnaire de file d'attente qmgr ;
3. active est une file contenant les messages en cours de délivrance par qmgr ;
4. deferred contient les messages qui n'ont pas pu être délivrés (il y en a 10 dans notre exemple).

De plus, le répertoire /var/spool/postfix/defer contient les mails en attente plus longue et des répertoires qui sont « hachés » afin de ne pas avoir des répertoires contenant trop de fichiers : ainsi, le fichier 7B345AC0B1, par exemple, sera dans defer/7/B/7B345AC0B1.

La documentation HTML de la distribution postfix dispose d'un schéma présentant clairement les interactions entre les différentes files d'attente et les différents démons. C'est d'ailleurs de celui-ci que je me suis inspiré...

Le contenu de la file d'attente peut être consulté avec la commande mailq (les habitués de sendmail ne seront pas dépaysés...) : normalement celle-ci ne doit produire que les messages dont la délivrance n'a pas encore eu lieu (dans notre cas, les 10 messages contenus dans la file deferred).

3.4. Script d'activation du serveur

/etc/rc.d/init.d/postfix start | stop | status | restart

3.5. Lancement du serveur

Ainsi que nous venons de le dire, quel que soit le système utilisé, c'est un script qui lance le serveur /usr/sbin/postfix en lui passant le paramètre start. Celui-ci lance à son tour le serveur principal, /usr/libexec/postfix/master qui prend alors les choses en main et lancera les autres démons lorsque cela sera nécessaire. Ces derniers se termineront après avoir accompli leurs tâches ou après une certaine période d'inactivité. Seul, le démon de gestion de la file d'attente, /usr/libexec/postfix/qmgr reste en permanence en activité. Tout ceci peut se vérifier par une simple commande ps (ici sous FreeBSD, d'où la présence de ces serveurs sous /usr/local/) :

```
% ps axf

...
583  ?  S    0:00 /usr/local/libexec/postfix/master
584  ?  S    0:00  \_ pickup -t fifo -c
585  ?  S    0:00  \_ qmgr -t fifo -u -c
...
```

qui met en évidence la présence du démon master et le fait qu'il a lui-même lancé les démons pickup et qmgr (la commande ps utilisée ici est celle de GNU, l'option -f n'a pas la même signification avec un ps BSD).

pickup est responsable de la récupération des courriers locaux : comme nous l'écrivions plus haut, pour des raisons de compatibilité, postfix utilise un programme nommé /usr/sbin/sendmail (qui n'est pas le programme sendmail bien connu, mais un homonyme). Ce programme est utilisé pour déposer les courriers locaux dans la file d'attente maildrop : tous les courriers qui sont postés par tous les utilisateurs sont déposés dans cette file.

pickup les récupère alors et les passe au démon cleanup qui remplira les en-têtes manquants, gérera les enveloppes des messages et les déposera enfin dans une autre file d'attente, nommée incoming. Puis cleanup avertira le gestionnaire de file d'attente, qmgr, qu'un nouveau courrier est arrivé.

qmgr s'occupera alors de délivrer le courrier dans les boîtes aux lettres de leurs destinataires et de gérer les erreurs.

Nous verrons plus loin les autres démons entrant en jeu dans la délivrance du courrier. Passons maintenant à une configuration minimum pour tester localement postfix.

3.6. Configuration minimale

Notre but, ici, est d'arriver à poster et recevoir du courrier en local : par exemple, root doit pouvoir poster un message à l'utilisateur babe. Ce dernier doit pouvoir récupérer le message, le lire et répondre à root. Pour simplifier, nous utiliserons le programme canonique mail.

Nous supposerons que notre machine s'appelle alex et que notre domaine s'appelle linux-france.org. Vérifions tout de suite que c'est le cas :

```
% hostname
alex.linux-france.org
```

Ainsi que nous l'avons déjà dit, la majeure partie du travail de configuration consiste à adapter le fichier /etc/postfix/main.cf (ou /usr/local/etc/postfix/main.cf) à nos besoins. Bien entendu, tout cela doit se faire sous le compte root.

La première chose à faire est de sauvegarder le fichier original :

```
# cp /etc/postfix/main.cf /etc/postfix/main.cf.0
```

Puis, chargez main.cf dans votre éditeur de texte favori. Normalement, un certain nombre d'options sont déjà en place. Certaines conviennent, d'autres non. Voici les lignes qui nous intéressent (les commentaires et les lignes non modifiées ont été supprimés) :

```
# INFORMATIONS SUR LES REPERTOIRES LOCAUX
```

```
queue_directory = /var/spool/postfix
command_directory = /usr/local/sbin
daemon_directory = /usr/local/libexec/postfix

# POSSESSION DES FILES D'ATTENTE ET DES PROCESSUS
mail_owner = postfix

# NOMS DE LA MACHINE ET DU DOMAINE
myhostname = alex.linux-france.org

# POUR L'ENVOI DU COURRIER
myorigin = $myhostname

# MODE DE TRANSPORT
default_transport = smtp

# GESTION DES ALIAS
alias_maps = hash:/etc/postfix/aliases
alias_database = hash:/etc/postfix/aliases

# DELIVRANCE DU COURRIER
mailbox_command = /usr/local/bin/procmail
```

Assurez-vous que toutes les autres possibilités pour ces lignes soient considérées comme des commentaires en les faisant précéder du caractère dièse (#) et ne modifiez pas les autres.

Avant de tester tout cela, détaillons rapidement les options choisies (pour des renseignements plus précis, reportez-vous aux pages de manuel et à la documentation fournie avec le programme).

La première section sert à spécifier les emplacements :

- /var/spool/postfix est le répertoire de base pour toutes les files d'attente de postfix, Lors de son premier lancement, postfix créera tous les sous-répertoires pour ses files sous ce répertoire ;
- /usr/local/sbin est le répertoire où se trouvent les commandes de postfix (les exécutables dont le nom commence par post, et sa version de sendmail) ;
- /usr/local/libexec/postfix est le répertoire contenant les démons de postfix : c'est là que se trouvent tous les programmes serveurs qu'il utilise.

La deuxième section précise qui est le propriétaire de la file d'attente et de la plupart des processus serveurs de postfix. Ici, nous avons conservé la proposition, après avoir créé l'utilisateur postfix. Voici son entrée dans notre fichier /etc/passwd :

```
postfix:x:101:101::/var/spool/postfix:/bin/false
```

Le 'x' dans la partie mot de passe vient du fait que nous utilisons les « shadow passwords ». Le groupe 101 correspond au groupe postfix, lui aussi créé pour l'occasion :

```
# grep 101 /etc/group
postfix:x:101:
```

La troisième section sert à indiquer le nom complet de notre machine.

La section suivante concerne l'envoi du courrier : elle permet de renseigner postfix sur la machine qui a posté. Pour le moment, nous considérerons que c'est ce que contient la variable

`$myhostname`.

Nous précisons ensuite le protocole utilisé pour l'acheminement du courrier. Pour l'instant, postfix ne reconnaît que smtp et uucp (en réalité, on peut créer des transports dans /etc/postfix/master.cf, ce qui permet de changer des paramètres en fonction de multiples critères. On peut ainsi dupliquer le transport smtp et en changer les caractéristiques en fonction des courriers entrants ou sortants ce qui est très souple. Toutefois, ne l'ayant pas pratiqué, je n'en dirais pas plus...).

La gestion des alias peut faire appel au fichier /etc/aliases utilisé par ses prédécesseurs mais nous préférons en utiliser un autre : /etc/postfix/aliases (ou /usr/local/etc/postfix/aliases). La commande man aliases vous renseignera en détail sur le format de ce fichier. Disons simplement que, comme son nom l'indique, il permet de définir des alias entre des noms de destinataires. Ainsi, par exemple, un serveur de news poste quotidiennement un rapport sur ses activités à l'utilisateur news (ou usenet). Supposons que babe soit l'administrateur des news sur alex : pour qu'il puisse recevoir ces messages, et si root est d'accord, bien entendu, il suffit d'indiquer que les destinataires news et usenet ont pour alias babe. Ceci est réalisé par l'ajout de la ligne suivante dans /etc/postfix/aliases :

```
news: babe
usenet: babe
```

Pour des raisons d'optimisation, postfix, comme ses prédécesseurs, demande à ce que ce fichier soit traité comme une base de données au format DBM ou DB. Pour générer ces formats, on utilise l'utilitaire /usr/sbin/postalias (qui, rappelons-le, est un lien vers /usr/local/sbin/postalias). Ma machine ne reconnaissant pas le format DBM, j'ai donc opté pour le second et produit la base à l'aide de la commande :

```
# postalias hash:/etc/postfix/aliases
```

qui a engendré le fichier /etc/postfix/aliases.db.

La section concernant la délivrance du courrier local indique ici que nous souhaitons utiliser procmail pour cette tâche. Tout autre programme ayant la même fonction peut convenir (deliver, par exemple),

mais procmail est le plus connu dans le monde Linux et FreeBSD. Cette section ne concerne que l'acheminement local du courrier, ie. son écriture dans les boîtes aux lettres des destinataires.

Test de la configuration locale

Après toute modification de l'un des fichiers que nous venons d'étudier, il faut demander à postfix de relire sa configuration :

```
# postfix reload
```

À l'aide de la commande ps ax, vérifiez la présence des démons master, pickup et qmgr. Si vous ne les voyez pas, c'est qu'il y a eu un problème : consultez les fichiers /var/log/mail.*pour tenter d'en rechercher la cause.

Si tout s'est bien passé, root va pouvoir envoyer un courrier à babe :

```
# mail babe -s test
premier test local
.
Cc:
#
```

Immédiatement, babe a dû recevoir ce courrier :

```
% mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/spool/mail/babe": 1 message 1 new
N  1 root@alex.linux-france.org.  Wed Jan 20 01:46  12/424   "test"
& 1
Message 1:
From root@alex.linux-france.org Wed Jan 20 01:46:10 1999
Delivered-To: babe@alex.linux-france.org
To: babe@alex.linux-france.org
Subject: test
Date: Wed, 20 Jan 1999 01:46:10 +0100 (CET)
From: root@alex.linux-france.org

premier test local
& d
& q
```

On notera la présence d'un champ Delivered-To: : il est ajouté par postfix afin d'éviter les boucles dans la délivrance du courrier et ne sera pas affiché par défaut avec la plupart des logiciels de lecture du courrier (en tous cas, avec Gnus et Netscape...).

Essayons encore : maintenant postez sous le compte babe un message à root et vite, très vite, faites ps axf. Sous Linux, vous devriez voir les lignes suivantes :

```
1271  ?  S    0:00 /usr/local/libexec/postfix/master
1272  ?  S    0:00  \_ pickup -t fifo -c
1273  ?  S    0:00  \_ qmgr -t fifo -u -c
1286  ?  S    0:00  \_ cleanup -t unix -u -c
1287  ?  S    0:00  \_ trivial-rewrite -n rewrite -t unix -u -c
1288  ?  S    0:00  \_ local -t unix
```

Assez rapidement, vous remarquerez, si vous faites la même commande, que cleanup et local se sont terminés, tandis que trivial-rewrite survit plus longtemps, puis se termine.

Tout ceci correspond ce que nous disions plus haut : pickup a récupéré le message et l'a passé à cleanup. trivial-rewrite s'est chargé de réécrire l'adresse en rajoutant le nom de la machine locale derrière le nom de l'utilisateur. local est le démon responsable de la délivrance locale du message dans la boîte aux lettres du destinataire. C'est à ce moment que le fichier des alias est pris en compte et, si le destinataire utilise un fichier ~/.forward, local le fait suivre à l'adresse indiquée. C'est local qui ajoute le champ Delivered-To: pour éviter un bouclage intempestif et c'est lui qui remplit le champ From de l'enveloppe (à ne pas confondre avec le champ From:...).

Pour finir cette partie, il ne vous reste plus qu'à essayer de faire la même chose avec vos logiciels de lecture de courrier favoris : cela ne devrait pas poser de problème puisque local délivre les messages à l'endroit où la plupart des logiciels s'attendent à les trouver.

3.7. Les logs

L'activité du serveur est prise en charge par syslogd. Vous trouverez les traces dans "/var/log/maillog".

3.8. Conclusion pour Postfix

Le produit est juste survolé, mais Postfix offre de nombreuses autres fonctions supplémentaires pour :

1. la gestion des listes
2. le traitement de l'anti-relayage et le SPAM
3. la réécriture d'adresse
4. filtrage par analyse (grep) des entêtes...

4. TP

Attention : avant de commencer vérifiez que vous n'avez aucun serveur de messagerie d'installé (sendmail, qmail, postfix...), sinon désinstallez ces packages.

4.1. Installation du serveur SMTP

Installez les packages de Postfix sur votre configuration. Configurez votre machine pour un service minimum (pas de liste, pas de réécriture d'adresse...). Vérifiez le bon démarrage du serveur.

4.2. Test de la configuration du serveur SMTP

Créez sur la machine locale un compte de test et utilisez le programme mail pour envoyer un message. Vérifiez sous ce compte que le message est bien arrivé.

4.3. Installation du serveur PostOFFICE Pop3

Installez le service pop3 dans /etc/inetd.conf ou dans /etc/xinetd.d de la façon suivante.

```
Exemple avec xinetd :
[mlx@uranus mlx]$ more /etc/xinetd.d/pop3s
# default: off
# description: 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   = yes
}
```

Dans inetd.conf décommentez la ligne. Dans xinetd, mettez la variable "disable à yes".

Relancez le service inetd ou xinetd.

4.4. Test du serveur Pop3

Vous allez réaliser l'opération à partir de la machine locale et d'une machine distante. La résolution de noms doit fonctionner, sinon utilisez les adresses IP.

1. Sur la machine locale qui est votre serveur SMTP et serveur POP3, configurez Nestcape Messenger ou kmail avec les paramètres suivants :

   ```
   Serveur smtp : Nom de votre serveur (Machine locale)
   Serveur POP :  Nom de votre serveur POP3 (Machine locale)
   Votre compte d'utilisateur
   Votre mot de passe
   ```

   Testez l'envoi et la réception de message.

2. Sur un client, configurez Nestcape Messenger ou kmail 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.

Installation d’un serveur PostgreSQL

Serveur WEB dynamique avec PostgreSQL et PHP

Alix MASCRET

Mode d’utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

1. Présentation

Accès à une base de données Postgres à partir d’un client WEB (Netscape ou autres)

On veut à partir d’un client « Web » comme Netscape (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.

2. 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).

2.1. 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.)

2.1.1. 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.

2.1.2. 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.

2.1.3. 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++

2.1.4. Les comptes utilisateurs :

Le compte administrateur de la base est par défaut " *postgres* "

il faut créer les comptes utilisateurs

Nobody, est le compte utilisé pour les accès anonymes via HTTP par exemple.

2.2. 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

destroydb [dbname]

destroydb [-h host] [-p port] [-i] [dbname]

Exemple *detroydb 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 " nobody ".

Utiliser la commande "select * from pg_user;" pour avoir la liste des utilisateurs.

Supprimer un utilisateur

destroyuser [username]

destroyuser [-h host] [-p port] [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]

Exemple: psql template1

Welcome to the POSTGRESQL interactive sql monitor:

Please read the file COPYRIGHT for copyright terms of POSTGRESQL

[PostgreSQL 6.5.1 on i686-pc-linux-gnu, compiled by gcc pgcc-2.91.66]

type ? for help on slash commands

type to quit

type or terminate with semicolon to execute query

You are currently connected to the database: template1

template1=>

template1=> \? / Obtenir les commandes de base */*

template1=\q

3. 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.

3.1. 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 .php3. Voir ci-dessous l'exemple " test.php3 " qui permet de vérifier le support de PHP par votre environnement.

Listing : test.php3

```
<?
 echo (" Test du module PHP ");
 phpinfo();
?>
```

3.2. 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.

4. Dialogue client et serveurs PHP, Apache et PostgreSQL

Voir schéma de la représentation en couches

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.

5. 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.php3" METHOD=post> // Ici le script qui sera exécuté
<textarea cols="100" rows="30"
name="c_SQL">Select * from phonebook ;</textarea></p>
<br>
```

```
 <INPUT TYPE="submit" VALUE="Search!">
 </form>
 </body>
 </html>
```

Le script associé : Page resultsql.php3

```
 '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);
 }
 ?>
```

6. TP

6.1. Présentation

Accès à une base de données POSTGRES 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 "fordemo.sql"
2. le document HTML "formsql.html"
3. le document php "resultsql.php2"

6.2. PostgreSQL

Connectez-vous en tant que *root.*

1 Préparation de la configuration

Installez les packages correspondant à PostgreSQL et à PHP à partir du CDROM s'ils ne sont pas déjà installés.

2 Configuration de Postgres

2.1 Postgres est installé. Le script de lancement est dans *« /etc/rc.d/init.d »*

Editez ce script et relevez :

l'emplacement et le nom du fichier dans lequel est placé le PID de PostgreSQL

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 »*

2.2 Il s'agit maintenant d'activer le service. Utilisez les commandes :

/etc/rc.d/init.d/postgresql stop

/etc/rc.d/init.d/postgresql start

Vérifiez le chargement de postgres dans la table des processus : *« ps aux | grep post »*

Attention, au lancement, le processus doit pouvoir écrire sur /tmp. Vérifiez les permissions. Si vous avez une erreur, vérifiez :

- 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

Vous devriez obtenir ceci :

$ *psql template1*

Welcome to the POSTGRESQL interactive sql monitor:

Please read the file COPYRIGHT for copyright terms of POSTGRESQL

type ? for help on slash commands

type to quit

type or terminate with semicolon to execute query

You are currently connected to the database: template1

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.

6.3. 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. « *nobody* » qui est utilisé pour les accès HTTP, « *TP1* » que vous utiliserez comme compte local.

3.1 Normalement *nobody* existe déjà, vous pouvez vérifier avec *« grep nobody /etc/passwd »*. Si ce n'est pas le cas, vous devrez créer un compte système pour nobody.

Attention : Le compte nobody est le compte utilisé par Apache. Si le nom du compte utilisateur est différent, vous devrez remplacer "nobody" par le nom de votre compte.

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 nobody et TP1. Attention aux réponses que vous mettrez car nobody 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 nobody

#passer en DBA (Data Base Administrator)

su postgres

$ *createuser nobody*

Enter user's postgres ID or RETURN to use unix user ID: 99 ->

Is user « nobody » allowed to create databases (y/n) *n*

Is user « nobody » allowed to add users? (y/n) *n*

createuser: nobody was successfully added

#c'est terminé

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 Nobody 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 nobody ;

\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

6.4. Serveur Apache et PHP

Installez le package mod_php*rpm (module php pour Apache

Démarrez le serveur Apache : */etc/rc.d/init.d/httpd restart*

Vérifiez que le serveur est bien actif et opérationnel.

Cherchez et relevez l'emplacement de stockage (Home Directory) des pages html d'Apache. En général dans « /home/httpd/html », sinon utilisez la commande *« find / -name html* ».

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.php3* suivant:

```
<? echo (« Test du module PHP »);
phpinfo();
?>
```

2 - Lancez Netscape (à partir de votre poste ou d'une autre machine sous linux ou win95) et tapez l'url « *http://@IP de votre PC/testphp.php3* » .

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,
- soit ce n'est pas le cas, il faut revoir la configuration.

6.5. 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.php3*) :

- le premier est une page HTML qui permet de saisir et « passer » des requêtes sql,
- le deuxième au format PHP3, met en forme le résultat de la requête.

1.2 Installez le support de postgres pour PHP, *mod_php3-pgsql-3.0.16-3mdk.i586.rpm* relancez Apache et refaites le test.

Copiez ces documents dans le répertoire où sont stockés les documents html du serveur.

Lancez ensuite Netscape. 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 le document resultsql.php3 afin de pouvoir réaliser des insertions dans la base. Les enregistrements à insérer seront saisies dans insert.html.

Réseau : Fichiers de confi guration et commandes de base

Alix MASCRET

Mode d'utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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 nomss

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/sysconfig/network-scripts. Voici des exemples pour une interface*lo* et*eth0*:

```
#interface lo
DEVICE=lo
IPADDR=127.0.0.1
NETMASK=255.0.0.0
NETWORK=127.0.0.0
BROADCAST=127.255.255.255
ONBOOT=yes
BOOTPROTO=none


#interface eth0
DEVICE=eth0
IPADDR=192.168.1.1       Adresse IP de l'interface
NETMASK=255.255.255.0    Masque utilisé
NETWORK=192.168.1.0      Adresse de réseau
BROADCAST=192.168.1.255  Adresse de diffusion
ONBOOT=yes               Activation au démarrage
BOOTPROTO=none           Configurer l'interface pour Bootp ou DHCP
```

3.

3.1. La commande ifconfig

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.

Cette manipulation amène à la réflexion suivante:

Il est possible de diminuer la taille des paquets qui transitent sur le réseau. Cela permet de garantir la bande passante si certains traitements, comme les transferts avec FTP, sont gourmands, mais charge la fonction de découpage et d'assemblage des paquets. Les temps de transmission sont également ralentis.

Cette manipulation est parfois utilisée par certains prestataires de service sur Internet. En diminuant le MTU des routeurs sur lesquels les clients se connectent, les prestataires diminuent la taille des paquets qui transitent. Ils créent une forme de multiplexage. Ils s'assurent, ainsi, que des clients ne monopolisent pas la bande passante. Cette opération est au prorata du confort d'utilisation, car les temps de réponses sont très dégradés et la charge des routeurs accrue.

L'autre solution consiste à augmenter la bande passante, ou bien trouver une solution pour lisser l'utilisation du canal en fonction des besoins.

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  98.482 ms 98.961 ms
```

```
 5  sfinx-paris.remote-gw.internext.fr (194.79.190.250)  100.180 ms  93.482 ms 134.686 ms
 6  Internetway.gix-paris.ft.NET (194.68.129.236)  98.471 ms  129.621 ms  99.433  ms
 7  513.HSSI0-513.BACK1.PAR1.inetway.NET (194.98.1.214)  137.196 ms  113.508 ms  117.050 ms
 8  602.HSSI6-602.BACK1.NAN1.inetway.NET (194.98.1.194)  101.129 ms  106.520 ms  107.828 ms
 9  FE6-0.BORD1.NAN1.inetway.NET (194.53.76.228)  105.110 ms  132.936 ms  108.04 6 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.

Éléments de cours sur TCP/IP

Alix MASCRET

Mode d'utilisation du serveur Unix/Linux. Environnement BTS Informatique première et deuxième année. Lycée Beaupeyrat - Limoges

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.

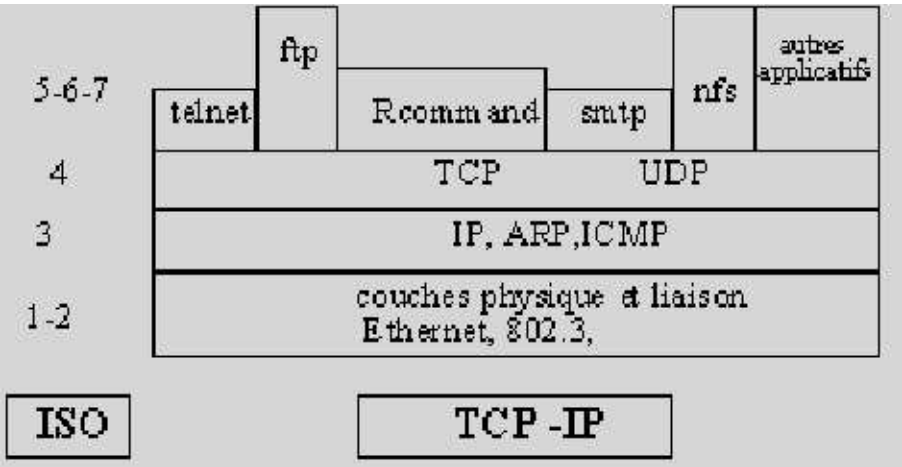
3. Les couches IP et TCP

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

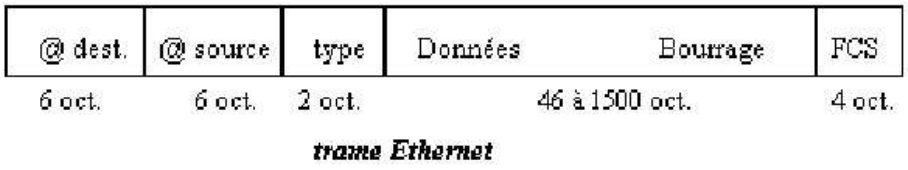
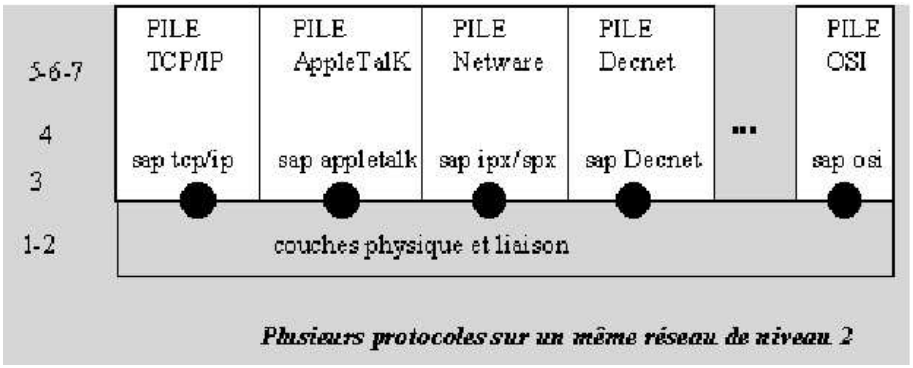


Figure 3. Les piles de protocoles



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

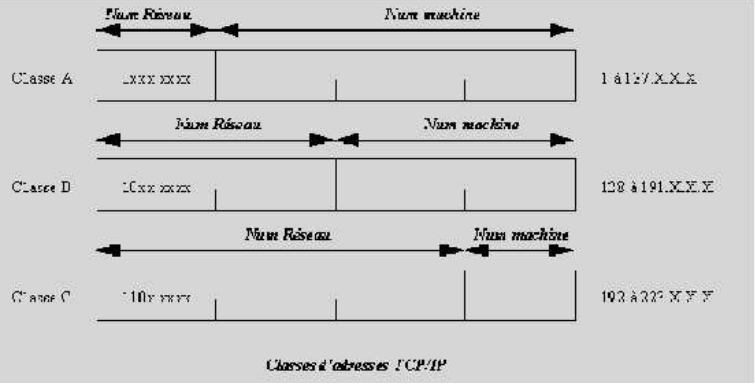
L'adresse IP est constituée d'un champ *numéro de réseau* (1, 2 ou 3 octets) et d'un champ *numéro de machine dans le réseau* (3, 2 ou 1 octets). L'adresse ip = adresse de réseau + adresse de machine.

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.

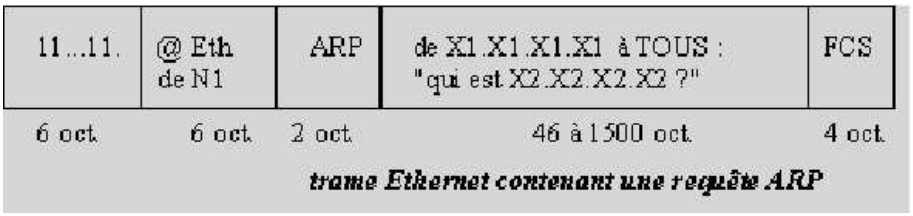
6. Le protocole ARP

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

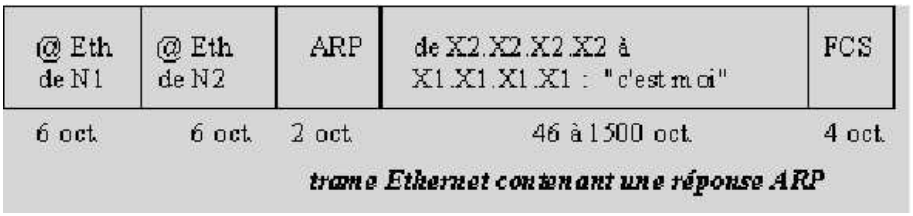
qui est X2.X2.X2.X2 ?

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 :

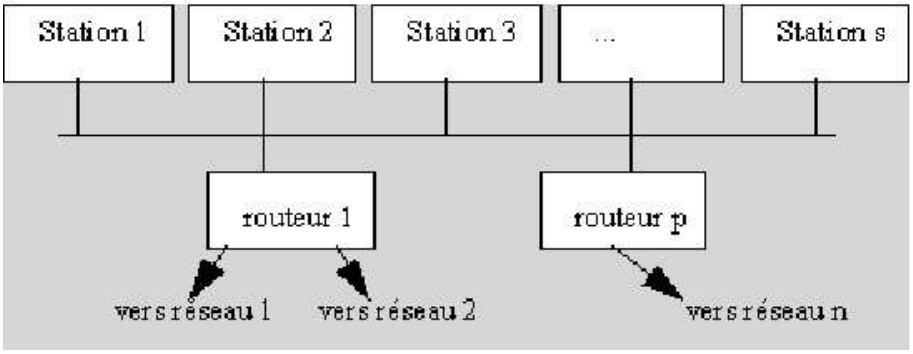
- Noms locaux : Sur chaque machine on crée un fichier qui contient la table de correspondance nom_machine --- @ip (par ex le fichier /etc/hosts sur un système Unix)
- Serveurs de noms : pour chaque domaine (par ex toubet.edu) une machine serveur de noms (serveur DNS) contient l'annuaire des machines 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

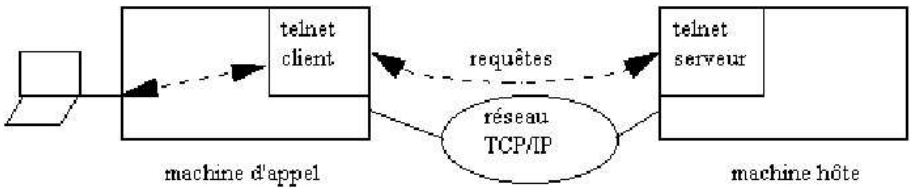
Les tables de routage peuvent être statiques dans le cas de réseaux simples, ou dynamiques dans le cas de réseaux maillés. Le protocole d'échange dynamique des tables IP sur un réseau local est *RIP* (Routing Information Protocol) ou le protocole OSPF.

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

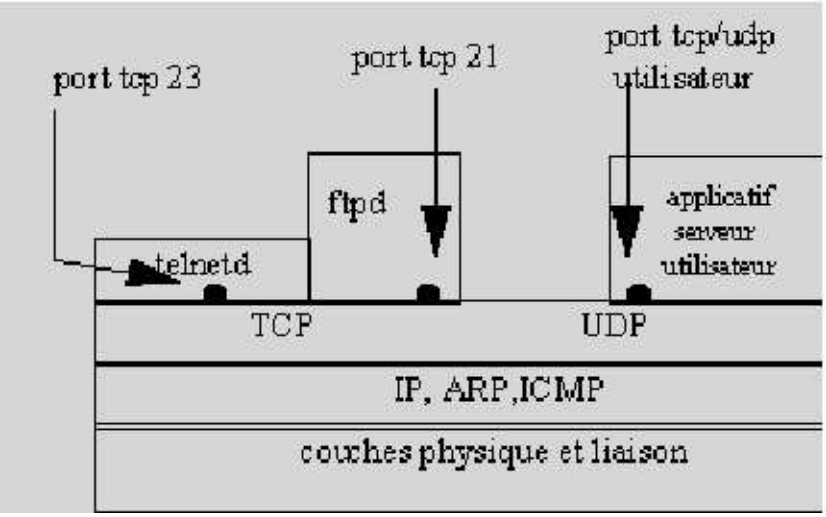


7.2. Adressage des applicatifs

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.

Utilisation des éditeurs joe et Emacs

Alix MASCRET

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
```