M3.23.4 Réseau Internet et services, Travaux dirigés et pratiques

Travaux dirigés et pratiques du troisième semestre

Jean-François Berdjugin
Jean-François Remm
Pierre-Alain Jacquot

M3.23.4 Réseau Internet et services, Travaux dirigés et pratiques: Travaux dirigés et pratiques du troisième semestre

par Jean-François Berdjugin, Jean-François Remm, et Pierre-Alain Jacquot

Publié le 20/10/08

Table des matières

Liste des illustrations

Liste des tableaux

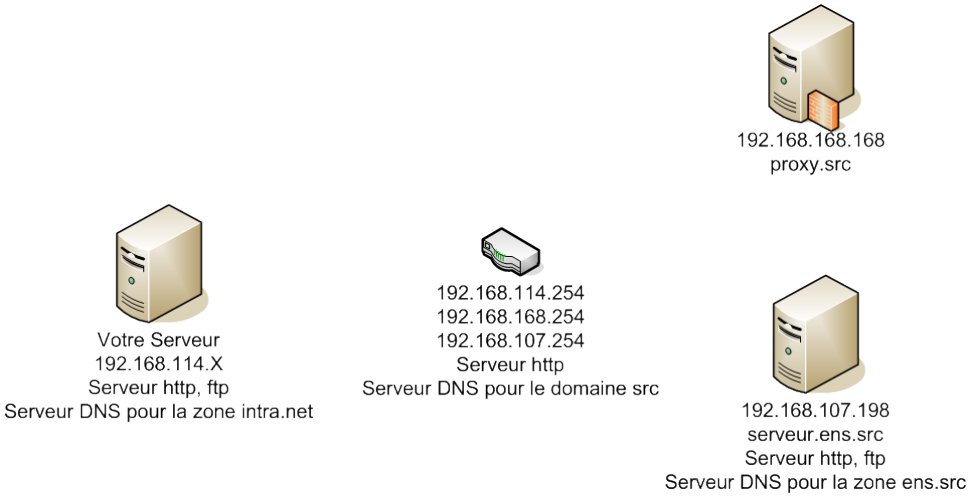
Liste des exemples

Chapitre 1. Présentation de l'architecture

Dans les TD (travaux dirigés) et les TP (travaux pratiques) qui vont suivre, vous aurez à gérer des services "Web" hébergés sur une seule machine. Cette machine gérera sur :

- Internet les services du domaine *S3X.ens.src* où X est le numéro de votre machine (de 1 à 15 et de 97 à 111),
- Intranet les services du domaine *.intra*.

Figure 1.1. Architecture réseaux.



Les TD sont à réaliser sous Windows et les TP sous Linux.

L'ensemble de l'administration de la première partie (jusqu'au déploiement d'applications Web) sera gérée depuis un compte utilisateur "standard". Vous disposez initialement sous Windows du compte *administrateur* avec comme mot de passe *admin*, sous Linux vous disposez du compte *user* avec comme mot de passe *adminad.*

Chapitre 2. Travaux dirigés

Préparation du poste de travail

Pour cette première partie les sites Web suivants ont été utilisés et peuvent vous être utiles :

- `http://technet.microsoft.com/fr-fr/default.aspx` Microsoft TechNet,
- `http://www.php.net/` le site de php (Hypertext Preprocessor),
- `http://www.mysql.org/` le site du SGBD (Système de gestion de bases de données) MySQL,
- `http://www.phpmyadmin.net/` le de site phpmyadmin une application php d'administration de mysql,
- `http://www.joomla.fr/` le site de joomla un système de gestion de contenu (CMS) développé en joomla,
- `http://www.dicofr.com` un dictionnaire en ligne.

Configuration réseau

Commencer par définir la configuration IP (Internet Protocol) de votre machine (adresse IP, masque, routeur par défaut). Vous utiliserez comme serveur DNS (Domain Name System) vous même.

Création d'un compte utilisateur et modification de la stratégie locale de sécurité.

Créer l'utilisateur *user* avec comme mot de passe *pass* en ligne de commande avec **net user** ou graphiquement :

1. Démarrer,
2. Outils d'administration,
3. Gestion de l'ordinateur,
4. Utilisateurs et groupes locaux.

C'est ce compte avec lequel nous ouvrirons prochainement une session.

L'utilisateur que nous venons de créer, ne peut arrêter le système, pour qu'il puisse le faire nous devons modifier la stratégie de sécurité locale.

Une stratégie de sécurité est un ensemble de paramètres de sécurité qui déterminent la sécurité d'un ordinateur. Elle permet de contrôler :

- Qui accède à l'ordinateur.
- À quelles ressources les utilisateurs sont autorisés à accéder sur votre ordinateur.
- Si les actions d'un utilisateur ou d'un groupe sont ou non enregistrées dans le journal des événements.

Si un ordinateur fait parti d'un domaine, la stratégie locale de sécurité est la dernière à s'appliquer. Ici votre PC ne fait pas partie d'un domaine donc pas de questions à se poser.

1. Démarrer,

2. Outils d'administration,
3. Stratégie de sécurité locale,
4. Stratégies locales,
5. Options de sécurité,
6. Arrêt Activé

De même vous devez permettre à *user* d'arrêter le système (Attribution des droits utilisateur, Arrêter le système). Commençons par permettre d'arrêter le système sans ouvrir une session.

Vous pouvez maintenant fermer la session Administrateur et ouvrir une session avec le compte *user*.

Comment depuis un compte utilisateur administrer votre Windows ?

Pour des raisons de sécurité, il est déraisonnable de travailler, de surfer avec un compte d'administrateur. Pour exécuter une commande, avec d'autre droits que ceux de l'utilisateur courant,vous avez deux possibilités :

- Majuscule + clique bouton droit + Exécuter en tant que ou
- la commande **runas**.

La commande runas permet à un utilisateur d'exécuter des outils et des programmes spécifiques avec des autorisations différentes de celles attribuées à l'ouverture de session, par exemple :

Exemple 2.1. Exemples d'utilisation de runas.

runas /user:companydomain\domainadmin "mmc %windir%\system32\compmgmt.msc" Lance la console de gestion de l'ordinateur avec les droits de l'utilisateur choisi (ici companydomain\domainadmin).

runas /user:Administrateur "desk.cpl" Afficher les propriétés d'affichage en tant qu'administrateur.

runas /user:Administrateur cmd Lance une invite de commande en tant qu'administrateur.

start "" /b "%ProgramFiles%\Internet Explorer\iexplore.exe" file:///c: A exécuter depuis une ligne de commande pour lancer Internet Explorer 6 (IE).[1]

Astuce

L'option **\sevcred** permet de sauvegarder le mot de passe.

Mise en place du DNS

La gestion des domaines *S3X.ens.src.* (X de 1 à 15 et de 97 à 111) est réalisée par la machine*192.168.107.198*.

Les noms d'hôtes suivants sont gérés :

- www.S3X.ens.src.
- ftp.S3X.ens.src.

[1]Ne fonctionne pas avec IE7, il faut dans ce cas détruire l'instance courante et la relancer avec le privilège d'administrateur en utilisant le gestionnaire des tâches :-).

- secure.S3X.ens.src.

Vous devez par contre gérer le domaine *intra* qui contiendra les noms disponibles sur l'Intranet :

- *www.intra.*
- *user1.intra.*
- *user2.intra.*
- *user3.intra.*
- *user4.intra.*

Vous allez devoir créer la zone de recherche directe *.intra* et la zone de recherche inverse *114.168.192.in-addr.arpa.*

Votre serveur DNS utilisera comme redirecteur *192.168.114.254.*

Pour administrer votre serveur DNS vous devez vérifier que le service est installé : **sc query DNS** et le cas échéant installer le service.

Le serveur DNS peut être administré graphiquement via l'interface Windows ou via la commande *dnscmd*. Cette commande n'est pas une commande installée avec Windows mais elle est disponible pour installation sur le CD : **e:\Support\Tools\SUPTOOLS.MSI.** Le script (*.bat*) suivant permet de configurer votre serveur DNS :

```
dnscmd localhost /zonedelete intra /f
dnscmd localhost /zoneadd intra /primary
dnscmd localhost /recordadd intra @ SOA serveur.intra admin.intra 0 3600 600 86400 3600
dnscmd localhost /recorddelete intra @ NS /f
dnscmd localhost /recordadd intra @ NS serveur.intra
dnscmd localhost /recordadd intra serveur A 192.168.114.1
dnscmd localhost /recordadd intra www CNAME serveur.intra
For /L %%i in (1,1,3) do (
dnscmd localhost /recordadd intra user%%i CNAME serveur.intra
)
dnscmd localhost /ResetForwarders 192.168.114.254
```

Evidement le script doit être modifié avant exécution. Pour la recherche inverse vous pouvez créé un nouveau script inspiré du précédent, la zone n'est plus intra mais 114.168.192.in-addr.arpa, le SOA et le NS restent identique, les A et CNAME sont remplacé par des PTR.

Mise en place des sites Web

Nous allons mettre en place des sites Web :

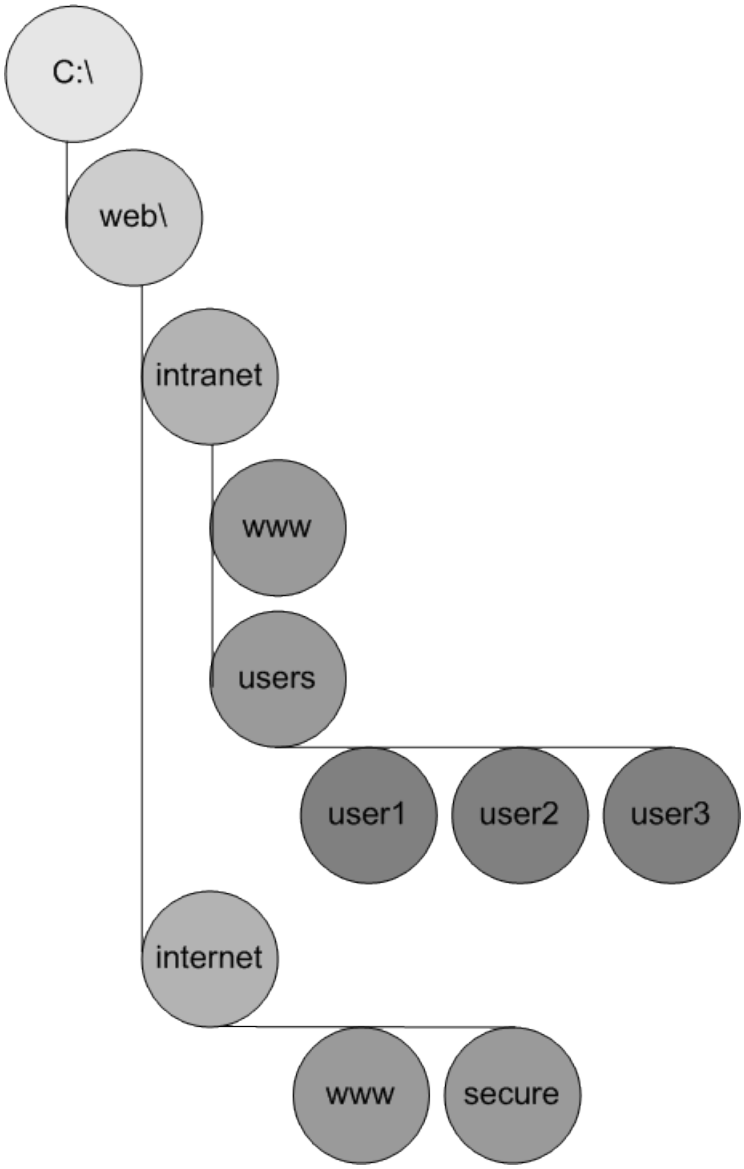
Tableau 2.1. Sites Web

Intranet	Internet
www.intra.	www.S3X.ens.src.
user1.intra.	secure.S3X.ens.src.
user2.intra.	
user3.intra.	

Ces sites Web seront ensuite complétés par un serveur FTP.

Les sites seront stockés physiquement comme suit :

Figure 2.1. Arborescence physique du serveur HTTP



Créer l'arborescence physique précédente et vérifier que le compte représentant pour IIS l'utilisateur anonyme puisse y accéder (celui-ci est dans le groupe utilisateurs).

Vous pouvez la créer graphiquement ou ligne de commande avec **mkdir**, **cd**, **dir**.

Vérifier si ce n'est fait que IIS est installé : **sc query | find /i "IIS"**.

Les commandes suivantes sont disponibles pour l'administration de IIS :

iisweb.vbs	Crée, supprime, arrête et liste les sites Web.
IIsFtp.vbs	Crée, supprime, arrête et liste le serveur FTP
iisvdir.vbs	Crée, supprime et affiche les répertoires virtuels de sites Web
IIsFtpdr.vbs	Crée, supprime et affiche les répertoires de "sites" FTP.
IIsCnfg.vbs	Importe et exporte la configuration IIS vers un fichier XML.
iisback.vbs	Sauvegarde et restaure les configuration IIS.

iisapp.vbs	Indique les PID des processus W3WP en cours d'exécution dans un pool particulier.
iisext.vbs	Configure les extensions des sites Web.
Iisreset	Redémarre les services internet.

Il est aussi possible de réaliser les opérations avec l'interface graphique :

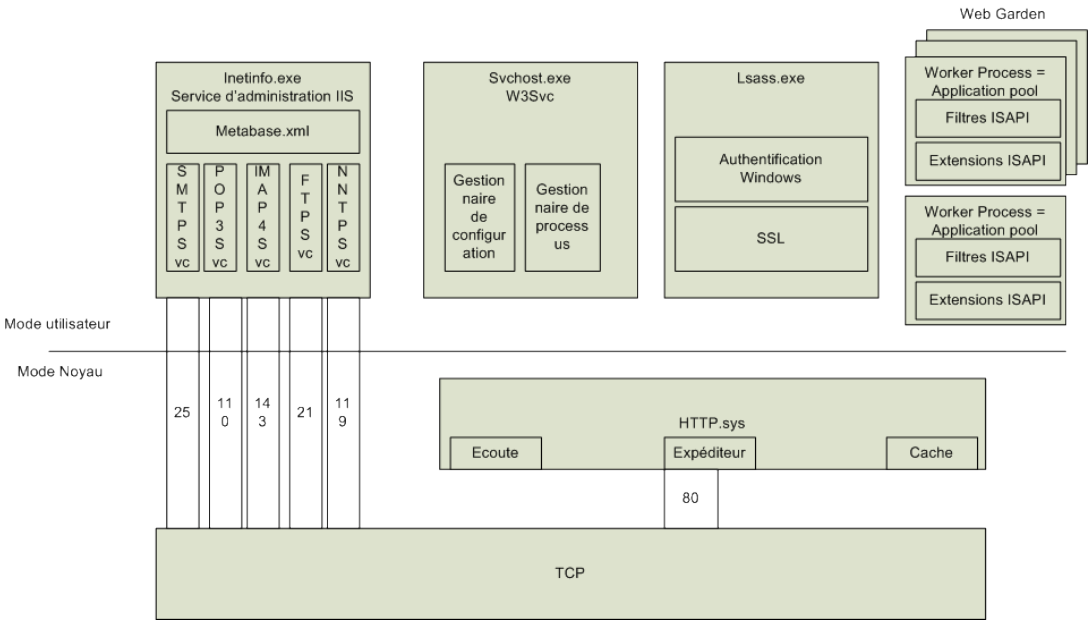
1. Démarrer
2. Panneau de configuration
3. Outils d'Administration
4. Gestion de l'ordinateur
5. Gestionnaire des services Internet (IIS)
6. Sites Web.

Architecture de IIS

Avant d'aller plus loin nous allons introduire sommairement l'architecture logiciel de IIS et la notion de *pool d'application*.

Les pools d'application définissent les paramètres de configuration et les frontières des applications. Un pool d'application est composé d'une ou plusieurs applications et lui ou leurs impose une exécution dans un contexte commun (exemple : même version d'asp.net pour tout le pool) et l'application des paramètres du pool (détection de panne, recyclage, surveillance du processeur,…) (`http://www.laboratoire-microsoft.org/articles/win/iis6/4/`).

Figure 2.2. Architecture de IIS



IIS est constitué d'un ensemble de composant :

HTTP.sys	Écoute et répartie les requêtes vers les applications concernées.
W3WP.exe	Représente un pool d'application.

Inetinfo.exe	Est un gestionnaire de configuration, d'application et de processus.
Svchost.exe (W3svc)	Service de publication sur le World Wide Web exécuté dans le contexte du processus Svchost.exe, un processus d'hôte générique pour les services implémentés dans des DLL.
lsass.exe	Processus servant aux authentifications sur le système Windows mais aussi à supporter Secure Socket Layer (SSL).
w3wp.exe	Processus de travail des applications, il permet de gérer plusieurs services Web réunis en un pool d'application. L'utilisation de pool d'application permet pour chaque pool de contrôler les ressources CPU, allocation de bande passante, l'arrêt, le redémarrage, ...

Vous aller créer deux pool d'applications, un qui sera utilisé pour l'intranet (*Intranet*) et l'autre pour Internet (*Internet*).

Les deux pools seront créés en utilisant l'assistant graphique. Au passage vous observerez que pour chaque pool vous pouvez définir :

- Une fonction de recyclage qui permet d'arrêter et de redémarrer les processus de travail utilisés par un pool d'application en cas de perte de mémoire ou de consommation anormale de temps CPU (Central Processing Unit),
- un ongle performance qui permet d'affiner la gestion du pool d'application,
- un onglet récupération et détection de panne qui permet de définir des tests sur l'état du pool d'application
- et enfin, un onglet identité qui indique le compte sous lequel les processus s'exécutent.

Sites simples

Nous allons dans un premier temps créer tous nos sites Web :

- www.intra,
- user1.intra,
- user2.intra,
- user3.intra
- secure.S3X.ens.src,
- www.S3X.ens.src.

La création manuelle avec l'interface graphique est possible mais peut-être longue. Il est préférable d'utiliser la ligne de commande.

Exemple 2.2. Exemple d'utilisation de iisweb

iisweb /create c:\web\intranet\www "www-intra" /d www.intra /ap intranetpermet de créer un site Web de qui dessert le nom d'hôte www.intra, dont le répertoire de base est c:\web\internet\www et qui fait parti du pool d'application intranet. Ce site sous Windows porte le nom de www-intra.

En vous inspirant de cet exemple créer les autres sites Web.

Astuce

Il peut être intéressant de créer un fichier *bat*.

Il vous faut aussi placer dans chaque site Web un fichier index.htm contenant le nom du site et enfin tester avec **iisweb /query** puis avec un client Web.

Vous pouvez aussi en utilisant les pools d'application arrêter, par exemple l'ensemble des sites de l'Intranet puis les relancer.

Secure Socket Layer (SSL)

Cette partie a été rédigé dans sa première version par Jean-Philippe Baudouin.

Introduction

L'objet de cette partie est de configurer un serveur Web sécurisé avec SSL. SSL est une couche supplémentaire permettant de combler deux objectifs dans le domaine des transactions sécurisées :

- certifier de façon unique et cryptée les interlocuteurs d'une transaction,
- sécuriser cette transaction.

Nous allons mettre en place un serveur délivré par la CA (l'autorité de certification) *thawte* (`http://www.thawte.fr/certificats-ssl-numeriques/essai-gratuit/index.html`).

SSL est actuellement le standard pour les transactions sécurisées sur Internet. L'utilisation de SSL permet l'authentification mutuelle du serveur et du client, le chiffrement et la vérification de l'intégrité des connexions.

SSL est l'abréviation de Secure Sockets Layer. Il faut prendre gare au sens dans lequel on emploie ce terme. L'IETF a spécifié un standard de sécurité appelé TLS (Transport Layer Secure) et c'est ce mode de sécurité qui est utilisé sur Internet, lui-même étant conçu à partir des spécificités SSL 3.0 dénies par la société Netscape. TLS en est actuellement à sa version 1.0.

SSL ne dépend pas des applications utilisées lors des transactions et s'applique sous les protocoles HTTP, FTP, Telnet, IMAP, SNMP, ... Clients et serveurs commencent par s'authentifier mutuellement, puis négocient une clé symétrique de session qui servira à assurer la confidentialité des transactions.

Les clés asymétriques utilisées lors des transactions SSL sont encapsulées dans des certificat X.509 version 3, générés par bon nombre d'autorités de certification, ou de PKI (Public Key Infrastructure).

Principe pour HTTP

SSL appelé par le client ou par le serveur HTTP à l'initialisation de la transaction s'intercale entre la couche HTTP et la couche session pour encoder le canal ouvert entre les sockets client et serveur.

Cependant les données propres avant encodage (cryptage) et après décodage (de-cryptage) restent au format HTTP (les transactions clients serveurs en HTTP sont conservées). Ceci se matérialise par l'URL serveur utilisée :`https://secure.S3X.ens.src/`

Lorsque qu'un client demande une URL de la forme `https://secure.S3X.ens.src/`, le serveur ouvre une socket sécurisée SSL en standard sur le port 443. Il envoie au client son certificat d'encodage, c'est-à-dire un fichier crypté contenant ses informations (société, nom du serveur, ..) ce qui permet de l'authentifier de façon unique (le client est certain de son interlocuteur). Le client accepte ce certificat et le stocke pour la durée de la transaction.

Le client génère une clé de session aléatoire qui sera la clé symétrique utilisée pour les transactions. Cette clé est envoyée cryptée avec la clé publique du serveur (donc l'envoi est considéré comme sûr).

Mise en place d'un certificat serveur

Un certificat SSL est unique pour chaque serveur WEB. Il lui est propre. Il consiste en un fichier texte. Les certificats SSL sont délivrés par des autorité de certification (CA Certification Authority) qui sont des organismes agrées par les gouvernements dont elles dépendent et habilitées à utiliser des systèmes cryptographiques.

La première étape est donc de générer une demande de certificat depuis le serveur WEB et d'envoyer cette demande à un certificateur.

Demande de certificat

Le certificat est propre au site, il doit donc demandé par le serveur à la CA est installé sur un site donné.

Nous allons sécuriser secure.S3X.ens.src pour ce faire, Aller dans

1. Gestionnaire des Services Internet
2. Propriétés du site Web
3. Sécurité du répertoire
4. Certificat de serveur pour générer la demande.

Vous pouvez utiliser IUT1 comme société et prendre SRC comme organisation. Le nom usuel à utiliser est secure.S3X.ens.src.

Obtention du certificat

Allez à l'adresse `http://www.thawte.fr` (Thawte est le certificateur que nous utiliserons), dans la rubrique "essais gratuits".

Étudiez les informations de sécurités envoyées par Thawte (double clique sur le cadenas). Examinez le certificat : comment se présente-t-il ? Qui est le certificateur ?

Utiliser les options par défaut. Copiez votre demande de certificat dans le champ prévu à cet effet. La demande de certificat est le texte contenu avec et entre les balises :

```
BEGIN ..
et
END ..
```

Envoyez le formulaire. La réponse est un certificat temporaire que vous allez installer sur votre serveur. Copiez le texte entre les balises dans un fichier texte sur votre disque, c'est le certificat.

Installation du certificat

Retourner dans

1. Gestionnaire des Services Internet,
2. Propriétés du site Web,
3. Sécurité du répertoire,
4. Certificat de serveur pour installer le certificat.

N'oubliez pas de configurer le port 443 sur votre serveur Web. C'est ce port qui est utilisé par défaut avec le butineur. N'autorisez l'accès à ce site qu'en utilisant le canal crypté (exiger un canal sécurisé).

Testez votre serveur sécurisé : `https://secure.S3X.ens.src`

Ré-examinez les informations sur la sécurité : Que dit le certificat ? Qui est le certificateur ? Qui est certifié ? Comme êtes-vous sur que la session SSL est lancée ?

Obliger maintenant le client à utiliser SSL en interdisant dans IIS les "connexion" non sécurisées.

Avec cet exemple, seul le serveur est certifié.

File Transfert Protocol (FTP)

Sur l'intranet nous souhaitions que des utilisateurs (*user1*, user2, user3) puissent déposer via FTP des données sur leurs sites Web via FTP.

Nous allons créer les trois comptes (*user1*, user2, user3) qui auront tous comme mot de passe *pass*, en utilisant la commande **net user**.

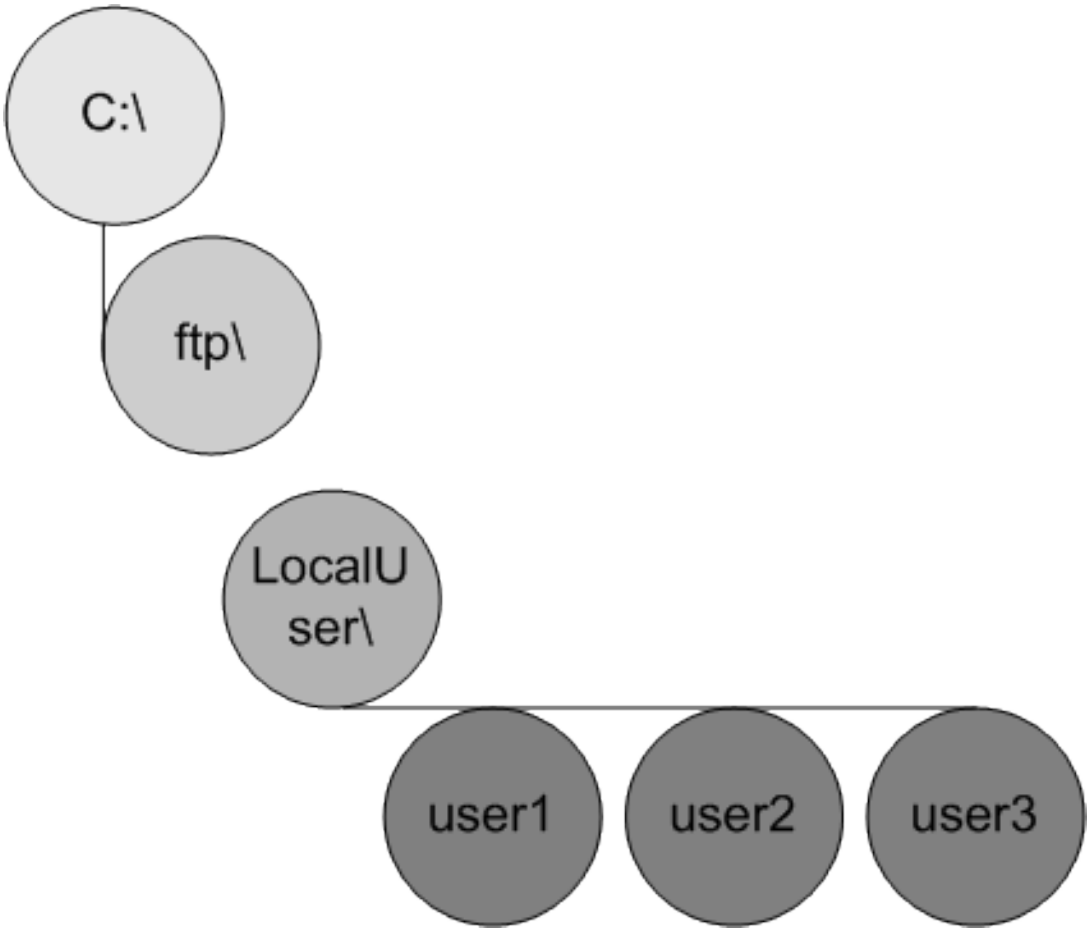
Vérifier si le service FTP est installé (**sc query | find /i "FTP"**) et si ce n'est pas le cas, installer le en utilisant (`appwiz.cpl`).

Un serveur FTP ne peut sur un port donné et une adresse IP donnée, servir qu'un seul site FTP. Contrairement au serveurs HTTP, la notion d'hôte virtuel n'existe pas. Nous allons donc commencer par arrêter le site FTP par défaut. Cette opération peut-être réalisée graphiquement ou en utilisant la commande **IIsFtp /stop**.

Les serveur FTP permettent *d'isoler* les utilisateurs. Dans un serveur isolé, un utilisateur ne peut accéder qu'à son propre répertoire.

Sous IIS une arborescence est imposé, les répertoires d'isolation doivent être, si la machine ne fait pas partie d'un domaine, positionner dans le répertoire `LocalUser` Lui même présent à la racine du site. Vous allez donc créer l'arborescence suivante :

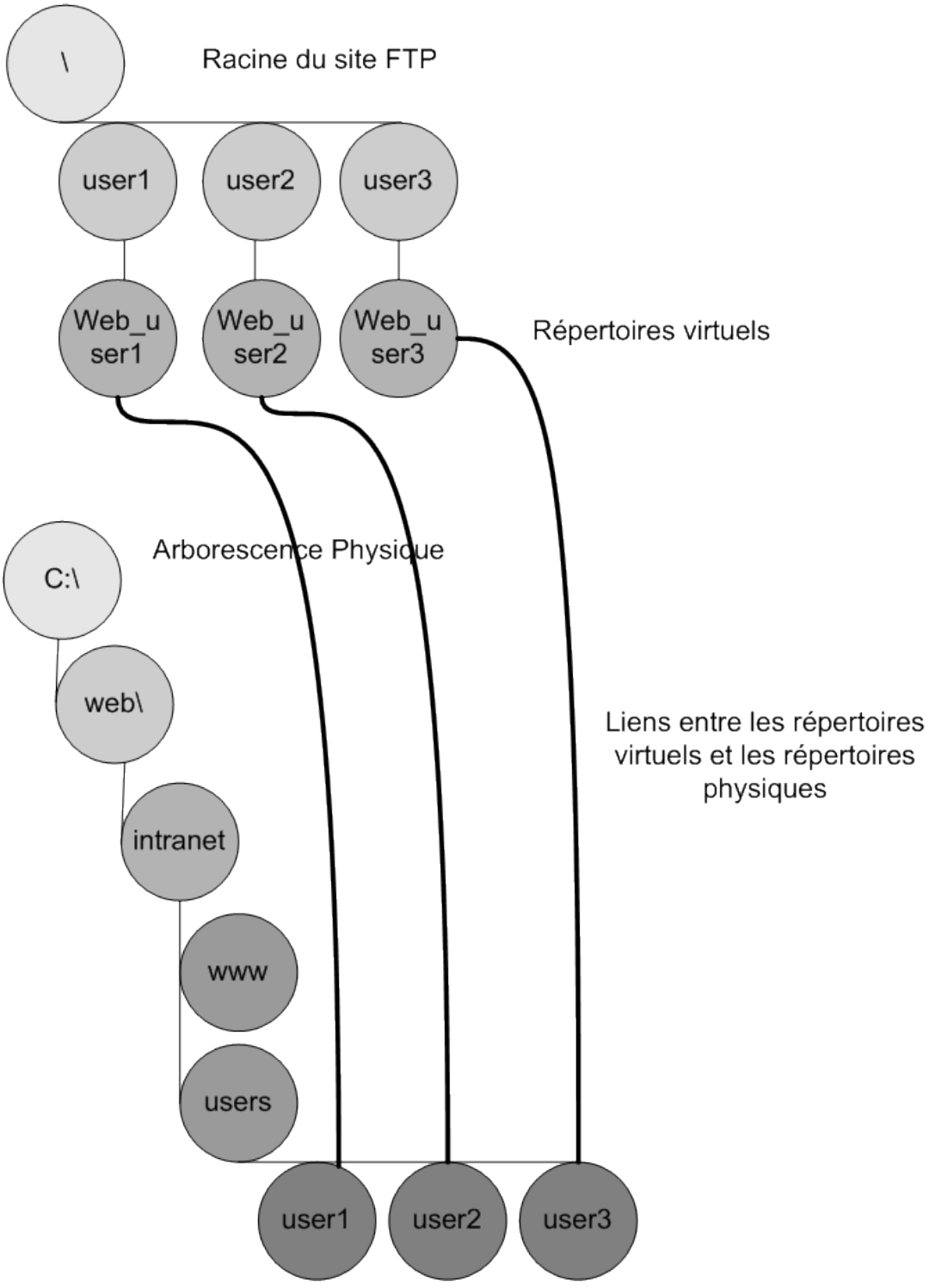
Figure 2.3. Arborescence physique du serveur FTP



En utilisant l'assistant graphique ou la commande (**iisftp /create**), créer le site FTP *users-intra* dont le répertoire de base est `c:\ftp`.

A ce stade, vous pouvez tester avec le FTP en ligne de commande ou avec un client graphique comme filezilla, mais notre serveur ne nous permet pas d'Uploader des fichiers vers les sites Web. Comment lier l'arborescence virtuelle FTP à l'arborescence physique du serveur Web : en utilisant des *répertoires virtuels*.

Figure 2.4. Lien en l'arborescence virtuelle du serveur FTP et l'arborescence physique du serveur HTTP



Créer dans le répertoire "FTP" de chaque utilisateur un répertoire virtuel pointant vers le site Web de l'utilisateur. Vous pouvez soit utiliser intégralement l'interface graphique ou la commande **iisftpdr /create**pour créer le répertoire virtuel et finir avec l'interface graphique pour le droit d'upload.

Il vous bien entendu demandé de tester en upload avec le client FTP de votre choix.
Il ne faut pas oublier les droits FTP, l'utilisateur doit pouvoir écrire (Uploder) dans son répertoire virtuel et qu'il doit aussi disposer des droits physiques sur le disque dur la commande cacls peut vous aider..

WebDAV (Web-based Distributed Authoring and Versioning)

Pour la publication sur `http://www.S3X.ens.src` nous allons utiliser *WebDAV* une extension (Internet Engineering Task Force) du protocole HTTP. WebDAV permet un travail collaboratif sur des fichiers distribués. Il est maintenant, particulièrement intéressant, car il permet de franchir les serveurs mandataires HTTP. Les clients WebDAV sont bien intégrés dans les OS (Operation System), sous Windows XP, les points de publication WebDAV sont accessible via les "Favoris réseau".

Point de publication WebDAV

WebDAV n'étant pas installé en standard, il vous faut vérifier qu'il est installé et activé dans le Gestionnaire des services Internet (IIS). Si il n'est pas installé, vous le trouverez dans :

1. Assistant Composants de Windows
2. Serveur d'applications
3. Services IIS
4. Services World Wide Web

Une fois WebDAV installé et autorisé comme extension de site Web, nous allons définir un point de publication pour *www.s3X.ens.src*.
Il est possible d'utiliser WebDAV sur SSL car WebDAV n'est qu'une extension HTTP. Ici nous ne le ferons pas car notre certificat est non signé par une autorité de certification, dont le certificat est présent sur notre machine, ce qui pose problème aux assistants Windows.

La procédure de publication est simple il faut créer un répertoire virtuel de nom WebDAV à l'endroit ou l'on souhaite publier. Créer un point de publication pour *www.S3X.ens.src*.

Accès à un répertoire WebDAV

Dans les favoris réseaux, créer un nouveau raccourcis ayant pour URL cette du site qui contient le répertoire virtuel "WebDAV".

Sous 2003 Server, il est impératif de démarrer le service "Webclient" pour permettre l'accès point de publication.

Sécurisation

Nous souhaitons que l'accès en upload soit réservé uniquement à l'utilisateur *user*, sur les deux sites. Pour cela et comme les fois précédantes, il faut positionner les DACL (Discretionary Access Control List) du système de fichier NTFS pour *user* ait les droits d'écriture sur les répertoires. Pour faire simple, vous pouvez donner un contrôle total à *user* et ne pas donner de droits aux autres utilisateurs.

Sur le point de publication, vous pouvez aussi affiner en fonction du contenu :

Lecture	Les utilisateurs peuvent afficher le contenu et les propriétés des répertoires ou des fichiers.
Écriture	Les utilisateurs peuvent modifier le contenu et les propriétés des répertoires ou des fichiers.
Accès à la source du script	Les utilisateurs peuvent accéder aux fichiers source. Si la case à cocher Lecture est activée, il est possible de lire la source. Si la case à cocher Écriture est activée, il est possible d'écrire dans la source. Accès à la source du

script inclut le code source des scripts. Cette case à cocher n'est pas disponible si aucune des cases à cocher Lecture ou Écriture n'est activée.

Important

Éviter pour des raisons de sécurité d'activer cette option.

Accès au journal	Exécuter les autorisations	
	Aucune	N'exécute aucun script ou exécutable sur le serveur.
	Scripts seulement	Exécute uniquement les scripts sur le serveur.
	Scripts et exécutables	Exécute les scripts et les exécutables sur le serveur

Nous reviendrons sur ces derniers points lors de l'utilisation PHP mais pour le moment, il vous faut tester les accès aux répertoire WebDAV. Vous devez demander une autorisation pour l'accès aux points de publication de votre voisin et faire en sorte que *user* soit le seul à pouvoir uploader.

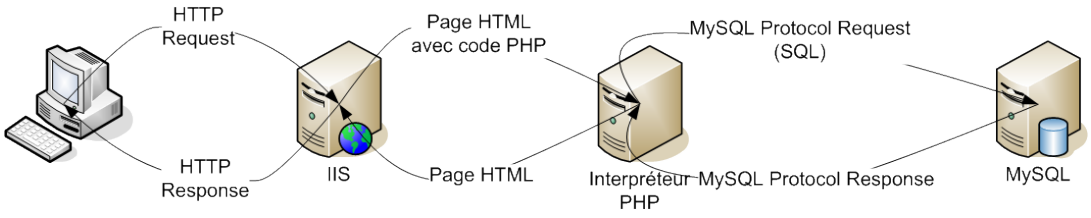
Installation du Système de gestion de base de données MySQL

Pour la partie qui suit vous pouvez ouvrir une session en "*Administrateur*", si les téléchargements des fichiers est un peu long une copie est disponible sur `ftp.ens.src`.

Dans ce TD nous utiliserons IIS (Internet Information Server) comme serveur HTTP (Hypertext Transfert Protocol), ce dernier pourra une fois configurer relayer les requêtes d'URL (Uniforme Ressource Locatoire) en .Php vers un interprétateur Php (Hypertext Preprocessor). Cet interprétateur communiquera avec le SGBD (Système de Gestion de Base de Donnée) MySQL. Cette architecture est qualifiée de trois tiers :

présentation	dialogue avec l'utilisateur (HTML, java script et le navigateur)
traitement métier des données	application (les scripts Php, le serveur HTTP et l'interprétateur Php)
Accès aux données	les requêtes et le SGBD.

Figure 2.5. Architecture Web trois tiers



Commençons par la fin et installons le SGBD.

Récupérer à l'URL suivante le fichier d'installation de la dernière version stable: `http://www.mysql.com` Choisissez une installation en tant que service Windows. Vérifier que le service est bien démarré : « sc query MySQL » ou graphiquement. En utilisant l'invite de commande fournie avec MySQL connectez vous au serveur puis tapez show databases. Si vous obtenez la liste des tables alors votre serveur est correctement installé.

Installation de Hypertext Preprocessor (Php)

Nous allons faire en sorte que nos site Web puissent contenir des sites Php.

Récupérer à l'URL suivante le fichier d'installation en version zip : `http://www.php.net`.

Déploiement d'applications Web

Nous allons installer l'interprétateur Php, puis PHPMyAdmin une interface Web d'administration de MySQL et enfin le CMS joomla.

interprétateur Php

L'installation de Php repose sur l'installation de DLL (Dynamic Link Library), sur la configuration de php.ini, le fichier de configuration de Php, et sur la configuration des filtres ISAPI (Internet Server Application Programming Interface) de IIS qui lui permettent de reconnaître et de traiter les fichiers en `.Php`.

Installation

IIS peut utiliser Php de deux manières :

cgi Common Gateway Interface, qui créé et exécute une instance php.exe à chaque exécution (lourd et non sécurisé)

ISAPI Internet Server Application Programming Interface, basée sur la technologie OLE (Object Linking and Embedding) qui permet le dialogue entre deux applications sous Windows. Avec cette approche Php ne sera chargé qu'une seule fois en mémoire.

Suivez la procédure suivante :

1. Décompresser l'archive dans un répertoire de votre choix (`c:\php5` chez moi).
2. Créer dans ce répertoire un répertoire `sessions` (`c:\php5\sessions`)
3. Copier `php5isapi.dll` (le lien entre IIS et Php) et `php5ts.dll` dans `c:\windows\system32\inetsrv`.
4. Copier `libmysql.dll` (pour pouvoir utiliser MySQL) dans `c:\windows\system32`.
5. Copier et renommer `php.ini-recommended` en php.ini (le fichier de configuration de Php) dans `c:\windows`.

Php.ini

Le fichier php.ini permet de configurer l'interpréteur Php. Il définit donc les fonctionnalités offertes par Php, comme par exemple, une limite d'upload, l'accès à une extension de traitement d'image (gd), l'accès depuis du code Php au SGBD MySQL, … Vous ne pourrez modifier le php.ini de votre fournisseur d'accès, il vous faudra donc inclure dans votre code les extensions dont vous avez besoin (dll).

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; Error handling and logging ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
error_reporting  =  E_ALL
;reporter tous les types d'erreurs
display_errors = on
;affiche les erreurs

;;;;;;;;;;;;;;;;;
; Data Handling ;
;;;;;;;;;;;;;;;;;
register_globals = Off
;à conserver à off pour des raisons de sécurité
;peut être modifié pour des scripts en php3

;;;;;;;;;;;;;;;;;;;;;;;;
```

```
; Paths and Directories ;
;;;;;;;;;;;;;;;;;;;;;;;;
include_path = ".;c:\php5\includes"
extension_dir = "c:\php5\ext"

;;;;;;;;;;;;;;;;;;;;
; Dynamic Extensions ;
;;;;;;;;;;;;;;;;;;;;
extension=php_mysql.dll
extension=php_gd2.dll

[Session]
session.save_path = "c:\php5\sessions"
session.auto_start = 0
```

Modifier votre `php.ini` pour que les valeurs de l'exemple précédant soient positionnées.

Configuration de IIS

Nous allons indiquer à IIS et ceux pour *tous* les sites que les fichier en `.Php` sont à transmettre à l'interpréteur Php:

1. Ajout de Php comme filtre Isapi : nom celui que vous voulez (moi php5), exécutable `c:\windows\system32\InetSrv\php5isapi.dll`.

2. Association de l'extension `.Php` au filtre Isapi. Aller dans répertoire de base configuration puis associer `.Php` à `c:\windows\system32\InetSrv\php5isapi.dll`.

3. Il nous faut maintenant autoriser l'extension .Php avec les extensions de site Web donner le nom que vous souhaitez à votre extension et associez la à `c:\windows\system32\InetSrv\php5isapi.dll`.

4. Rajouter dans les documents rechercher par IIS « `index.php` » en tête.

5. relancer IIS (**iisreset**).

Test

Créer un fichier index.php, placé dans le répertoire de base de *www.intra* et contenant :

```
<html>
<head>
</head>
<body>
Static <br/>
<?php
echo "dynamic";
phpinfo() ;
?>
</body>
</html>
```

Tester avec un navigateur. A ce stade nous disposons de tout ce dont nous avons besoin pour travailler mais nous pouvons choisir d'installer une interface d'administration pour « MySQL » comme « phpmyadmin ».

PHPMyAdmin

Récupérer à l'URL suivante le fichier d'installation : `http://www.phpmyadmin.net` Cette application Php se présente comme un ensemble de fichiers `.Php` et `.HTML`. Comment les rendre visibles depuis notre site Web ? Simplement avec un répertoire virtuel du site par défaut, ou d'un autre de vos sites si vous l'avez effacé, qui nous permet avec l'URL `http://localhost/phpmyadmin` d'accéder au `index.php` contenu dans le répertoire d'installation de phpmyadmin. Pour créer le répertoire virtuel vous pouvez le réaliser graphiquement ou avec la commande : **iisvdir /create « nom du cite » phpmyadmin «répertoire d'installation de phpmyadmin »**.

Vos problèmes devraient commencer, il y a bien évidement un fichier de configuration.

Un fichier pré-rempli, config.samples.inc.php existe, il vous faut le renommer en config.inc.php puis le modifier pour qu'il contienne :

```
<?php
...
$cfg['blowfish_secret'] = 'blabla';
...
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
$cfg['Servers'][$i]['compress'] = false;
/* Select mysqli if your server has it */
$cfg['Servers'][$i]['extension'] = 'mysql';
/* User for advanced features */
// $cfg['Servers'][$i]['controluser'] = 'root';
// $cfg['Servers'][$i]['controlpass'] = 'motdepassroot';
...
?>
```

motdepassroot est le mot de passe de l'administrateur (root) de MySQL.

Si il vous manque encore une chose ou deux à vous de modifier votre `php.ini`. En utilisant le fichier contenant *phpinfo()* vous pouvez vérifier la présence ou non de l'extension *MySQL et gd2*.

Une fois phpmyadmin opérationnel, il vous faut créer un compte *joomla* dans *MySQL* et une base de données *joomla*. Vous pouvez réaliser le tout en une étape en passant par le menu *privilèges*.

Joomla

Télécharger joomla à l'URL suivante `http://www.joomla.org` et déployé le à l'URL `http://localhost/joomla`.

Redirection et réécriture d'URL

Le protocole HTTP est riche, il permet des redirections :

HTTP 301 Permanent Redirect

HTTP 307 Temporary Redirect

HTTP 302 Moved Temporarily

Faites en sorte que la consultation du site `http://www.S3X.ens.src` conduise au site `https://secure.S3X.ens.src`.
Si vous n'être pas administrateur, une redirection en java script ou avec un script serveur reste possible.

La réécriture d'URL ne fait partie du protocole hotte et ne répond pas au problème du déplacement de site. Elle est spécifique à chaque serveur Web et répond au problème de l'accès à des pages dynamiques par les utilisateur ou les moteurs de recherche. Comment transformer `http://www.monsite.com/ruc.php?promo=2&year=2007` en `http://www.monsite.com/promo2/2007` ou `http://www.monsite.com/2-2007.php` ?

IIS est faiblement doté pour la réécriture, nous pouvons soit utiliser les fonctions internes, soit faire appel à un filtre ISAPI, soit utiliser l'architecture *asp.net 2.0* et l'URL mapping.

Fonctions internes

Dans les propriétés du répertoire de base, vous trouvez l'URL de redirection qui accepte les options suivantes :

Tableau 2.2. Règles de réécriture

Variable	Description	Par exemple
$S	Redirection de script	La chaîne " nombre = 1 " Est mappée en l'URL de destination. www.free.fr/nombre=1 devient
P to P	Transmet les paramètres de l'URL d'origine.	
$Q	Comme P mais avec pris en compte du point d'interrogation	La chaîne " ? nombre = 1 " est mappé en l'URL de destination.
$V	Transmet l'adresse URL demandée, sans le nom de serveur.	
de $0 à $9	Transmet la partie de URL demandée qui correspond aux caractères génériques indiqués.	
!	Ne redirigez pas.	

Bref par très explicite, tester une redirection avec `http://www.free.fr/$P` par exemple.

Filtre ISAPI

Vous trouverez à l'URL suivante `http://www.helicontech.com/`, la version gratuite du filtre ISAPI : *ISAPI_Rewrite*.

Une fois installés vous devez avoir les fichers suivants :

httpd.ini	fichier contenant les expressions régulières,
httpd.parse.errors	le fichier contenant les erreurs rencontrées,
ISAPI_Rewrite.dll	la DLL qui effectue la transformation
RXTest.exe	une interface graphique de teste des expressions régulières.

Essayer de transformer tous les fichiers .php en .php5 en utilisant la directive : **RewriteRule URL_affichée URL_chargée [Options]**.

Chapitre 3. Travaux pratiques

Dans ce chapitre, nous allons réaliser les services offerts dans le chapitre 1, mais cette fois ci, avec une solution Linux.

Cette année, la distributon choisie est la *Ubuntu 8.04 Hardy Heron* , téléchargeable gratuitement à l'URL : `http://www.ubuntu.net/`. Vous pouvez la tester, chez vous, sans risque, avec la version *Desktop* qui est utilisable en "live CD" et qui par conséquent peut-être utilisée sans installation sur le disque dur.

Préparation du poste de travail et découverte de Linux.

Dans cette partie, la découverte de l'auronomie fait partie de la difficulté. Les sites suivants vont vous être utils :

`https://help.ubuntu.com/`	le sitde de la documentation ubuntu en anglais,
`http://doc.ubuntu-fr.org/`	Le site de la documentation ubuntu en français,
`http://www.apache.org`	le site de la fondation apache
`http://www.isc.org`	le site de bind le serveur DNS de l'université de Berkley.

Avant de configurer notre ordinateur, nous allons apprendre à changer d'utilisateur.

Comment depuis un compte utilisateur administrer votre Linux.

La philosophie *ubuntu* est un peu particulière, la *ubuntu* fait partie de la famille *Debian*, le *root* (administrateur) n'a pas de mot de passe et il existe des utilisateurs avec pourvoir dont fait parti celui que vous avez utilisé (*user*) pour vous logguer. Graphiquement lorsque vous utilisez un utilitaire de configurartion système, le mot de passe de l'utilisateur avec pouvoir utilisé est demandé. Il est aussi possible d'utiliser la ligne de commande :

Exemple 3.1. Exemples de la commande sudo

Voici quelques utilisations de la commande **sudo**

sudo su	permet de changer l'utilisateur courant du terminal pour devenir *root*,
sudo gedit	permet de lancer un editeur avec des droits de *root*
sudo gnome-terminal	permet de lancer un autre terminal avec des droits de *root*,
sudo nautilus	permet de lancer le navigateur avec des droits de *root*

Pour certaines applications graphiques vous devez utiliser *gksudo*.

Astuce

Comme pour toutes les commandes vous pouvez obtenir de l'aide avec **man** ou avec le nom de la commande suivit de **--help**.

Fichiers de configuration et fichiers de log

Sous linux tout est fichier, que ce soit un document ou un périphérique, la plus part des fichiers sont des fichiers textes donc éditables. Parmi les fichiers qui nous intérressent, nous trouvons ceux de configuration et ceux de log. La notion de *base de registre n'existe pas* .

Fichiers de configuration

Sous linux vous trouverez les fichiers de configuration sous `/etc`.

Pour tester, ajouter une entrée dans le fichier `/etc/hosts` associant à votre adresse IP le nom d'hôte *test.intra*, puis tester avec **ping**.

De même expliquer ce que contient le fichier : `/etc/network/interfaces`.

Fichiers de log

Le système, les applications et les serveurs enregistrent des informations dans des fichiers qualifiés de *log*.

Les logs systèmes se trouvent sous `/var/log`, le fichier `/var/log/messages` et le fichier `/var/log/daemon` sont les principeaux. Le fichier `/var/log/dmesg` aussi accessible via la commande **dmseg** correspond aux messages du noyau.
Il peut être interressant de lancer dans un terminal la commande tail -f fichier_de_log. Cette commande permet un affichage continu et réactualisé.

Les logs des applications utilisateurs se trouvent sous leur *home directory* (`/home/user`). Vous pouvez par exemple observer le fichier `.bash_history`, par convention les fichiers cachés commencent par un point.

Commandes indispensables

Il est des commandes inévitables sous Linux.

Pour les fichiers nous trouvons :

- **cd**,
- **ls**,
- **mkdir**,
- **rm**,
- **mv**,
- **cp**,
- **ln**,
- **chmod**,
- **chown**,
- **tail**.

En vous aidant du man comprenez à quoi, elles peuvent bien servir.

Gestionnaire de paquet

Sous *Linux* il est possible d'ajouter de nouvelles applications soit par compilation, souvant en Cou C++ (utilisation de **gcc** avec **make** et **make install**), soit en utilisant un gestionnaire de paquet.

Il n'existe pas d'équivalent sous *Windows*. Sous la *Ubuntu* nous utiliserons APT (Advanced Packaging Tool), un sytème complet et avancé de gestion de paquets, permettant une recherche facile et efficace, une installation simple et une désinstallation propre de logiciels et utilitaires. Il permet aussi de facilement tenir à jour votre distribution

Ubuntu avec les paquets en versions les plus récentes et de passer à une nouvelle version de Ubuntu, lorsque celle-ci sort.

Nous utiliserons APT soit graphiquement avec *le gestionnaire de paquet synaptic* et *le gestionnaire de mise à jour*, soit en ligne de commande.

Les outils de lignes de commande indispensables sont : **apt-get** et **apt-file**.

apt-file est utilisée avec les options **update** pour mettre les fichiers à jour et avec **search** pour savoir si un paquet est installé et quels sont ses fichiers. Cette dernière option est particulièrement intéressente car vous pouvez savoir où sont et quels sont les fichiers. Elle peut-être complétée par l'utilisation de l'option **list**.

apt-get est utilisée avec les options **upgrade** pour mettre les fichiers à jour et avec **install** pour installer un nouveau paquet.

Le client DHCP étant installé trouvez les fichiers qui composent le paquet *dhcp3-client* en utilisant *apt-file* et éventuellement *apt-get*.
Si une commande est un peu longue, rien ne vous empêche de lire la suite du sujet.

Comment lancer les services ?

Sous linux, il existe trois manières de lancer les services : à la main, au moment du boot, en utilisant un *super daemon*.

Runlevels

Le premier processus à s'exécuter est *init* (son numéro de processus ou PID vaut 1) . Son rôle est de lancer les processus nécessaires au fonctionnement du système. L'exécution de init est paramétrée par `/etc/events.d` qui utilise les fichiers de configuration pésents dans `/etc/rcX.d` et `/etc/init.d`.

Les *runlevels* ou "niveaux d'exécution", correspondent aux services qui vont être lancés au démarrage de la machine.

Vous pouvez tester avec **init 1**.

Les script permettant de lancer les daemons sont aussi accessibles via **/etc/init.d nom_daemon** . Vous pouvez tester en arretant puis en relancant *networking*.

Le super daemon xinetd ou inetd

INternET Daemon ou inetd est un de servrvice un peu particulier. Sur les systèmes récents, comme la *ubuntu*, il est utilisé conjointement avec xinetd qui a l'avantage d'être configurable plus finement. Le principe de fonctionnement est le même, seule la conguration change. Inetd (ou xinetd) est lancé au démarrage. Il écoute sur certains ports. Quand une connexion est demandée sur un port, il détermine à quel service (l'ensemble des services connus est dans `/etc/services`) correspond le port et appelle le programme adéquat pour traiter la requête. xinetd permet de démarrer, à la demande, des démons réseaux en réduisant la charge de la machine. La conguration de ce super démon est définie dans `/etc/xinetd.conf`.

Configuration réseau

Il est possible de définir une configuration IP en utilisant des fichiers comme `/etc/resolv.conf` pour le client DNS et `/etc/network/interface` pour la configuration IP. Le problème est que ces fichiers peuvent être modifiés par des assistants graphiques dans que vous ne vous en rendiez compte. Nous allons donc passer directement par l'interface d'administration :

1. Système

2. Administration

3. Réseau

Astuce

Je vous conseille de cocher et de décocher la case activé pour que la modification soit bien prise en compte.

Mise en place du DNS

Nous allons reproduire ce que nous avons avec windows à savoir gérer les zones : *intra* et *114.168.192.in-addr.arpa*.

Nous allons successivement :

1. configurer un client d'un serveur de résolution de noms,
2. installer bind (un serveur de nom),
3. configurer un serveur de nom.

Principe du DNS

Le système de noms de domaine (Domain Name System) est avant tout la base de donnée distribuée du nommage hiérarchique des hôtes de l'Internet. Cette base est constitué de fichiers textes --dont le format est fixé et mis en place localement sur un serveur et accessible via un mécanisme de client serveur. La partie serveur du système est assuré par des programmes appelés serveurs de noms. La plus populaire mise en oeuvre est BIND (Berkley Internet Name Domain), actuellement maintenu par l'ISC (Internet Software Consortium). Les utilisateurs accèdent aux serveurs de noms par des programmes appelés *solveurs* (*resolver*). Pour faire correspondre une adresse ip à un nom, un programme appelle le *solveur* et lui passe le nom de l'hôte recherché en paramètre (ex : www.zone.fr). Le solveur envoie un paquet UDP (User Datagram Protocol) au serveur DNS configuré qui peut traiter la requête de quatre façons :

- Le nom demandé figure dans sa table et il donne la réponse.
- Le nom ne figure pas dans sa table. S'il connaît un serveur de zone.fr, il s'adresse directement à lui, sinon, c'est un serveur de fr, voire de la zone racine qui est sollicité. Nous supposons qu'il interroge l'un des serveurs de fr qui fait suivre la requête à l'un des serveur de zone.fr, obtient la réponse et la retransmet : requête récursive. C'est le mode le plus ``agressif''.
- Le nom ne figure pas dans sa table. La recherche (dans ses connaissance) du serveur hiérarchiquement le plus proche du nom recherché suit le même principe que la requête récursive (nous supposons ici que le premier serveur contacté est un serveur de la racine). Ce serveur sollicité n'enchaîne pas les requêtes, mais se contente d'envoyer la liste des serveur de la zone fr. Le serveur DNS local contacte alors un des serveurs de fr pour obtenir la liste des serveur de zone.fr avant d'en contacter un et d'obtenir sa réponse : requête itérative.
- Le nom ne figure pas dans sa table et le serveur est configuré pour rediriger les requêtes qu'il ne peut traiter vers un *redirecteur* (*forwarder*) qui essayera de lui renvoyer la réponse.

L'adresse ip ainsi obtenue est renvoyée au solveur qui la renvoie à l'appelant.

Domaine

La structure de la base de donnée est arborescente avec comme racine.et des domaines de haut niveau (top level domain). Chaque domaine est désigné par le chemin à suivre depuis la racine.

Délégation

Pour créer un nouveau domaine, on doit avoir l'autorisation de domaine parent (ex : ujf-grenoble.fr autorisé par le gestionnaires de fr). A l'opposé, on peut créer des sous-domaine de son propre domaine. (ex: ujf-grenoble.fr

pourrait créer src.ujf-grenoble.fr). Ce transfert de l'autorité de la gestion d'un sous-domaine s'appelle la délégation de zone.

Domaine vs zone

Pour un domaine non découpé en sous-domaine, la notion de zone et de domaine est identique. Pour des domaines découpés en sous-domaines, la zone contiendra essentiellement des informations de délégation aux sous-domaines, alors que le domaine recouvre la zone ``principale'' et les zones délégués.

Résolution inverse

Le problème de la résolution d'un nom vers un adresse étant résolu, il reste à savoir faire le contraire, c'est à dire passer d'un nom à une adresse. L'idée qui a été retenue est toute simple : dans un domaine particulier appelé in-addr.arpa, on utilise simplement des adresses comme nom des sous-domaines.

Rôle d'un serveur de nom

Le serveur de DNS a donc un rôle double :

- Il est l'interlocuteur du solveur pour diligenter une résolution de nom,
- il est de plus une partie des ressources (sur ses données).

Configuration du solveur (resolver)

Un hôte utilisant un solveur est appelé client. C'est à la configuration de cette partie cliente, à l'image de ce que vous pouvez faire quand vous avez un abonnement chez un fournisseur d'accès internet, qu'est consacré cette partie du tp.

Explication sur les fichiers de configuration

Fichier /etc/host.conf

e fichier /etc/host.conf contient la configuration du solveur de nom à l'aide de quelques directives (nous ne les listons pas toutes) :

`order` ordre de résolution avec comme valeurs possibles hosts, bind et nis. Un directive comme : `order hosts bind` signifie que l'on recherche d'abord à faire la résolution de nom en consultant le fichier local /etc/hosts avant de contacter un serveur DNS, si cette recherche s'avère infructueuse.

`multi` Les valeurs valides sont on et off. Si cette option est sur on, la bibliothèqe resolv+ renverra toutes les adresses valides pour un hôte apparaissant dans le fichier /etc/hosts plutôt que de ne renvoyer que la première. Par défaut elle est sur off car cela peut causer des dégradations sensibles des performances sur les sites ayant un gros fichier hosts.

Fichier /etc/hosts

C'est le fichier qui contient les hôtes locaux. Il évite la mise en place d'un serveur de nom, mais la mise à jour de ce fichier est rapidement problématique pour un réseaux local évolutif.

```
127.0.0.1       localhost
127.0.1.1       PC-JUB.maison.org       PC-JUB
192.168.114.1   test.intra

# The following lines are desirable for IPv6 capable hosts
::1     ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
```

```
ff02::3 ip6-allhosts
```

Fichier /etc/resolv.conf

Le comportement du solveur est configuré dans le fichier /etc/resolv.conf à l'aide de quelques directives dont voici un extrait :

domain — domaine par défaut. ex : si vous avez une entrée domain tp.net cela signifie que l'hôte local est dans le domaine tp.net Cette directive peut être omise au profit des informations qui figurent dans le fichier /etc/hosts.

search — cette directive permet de fixer la liste de recherche des domaines (jusqu'à 6) à parcourir. Ex : si vous avez une entrée search dom1.tp.net dom2.tp.net lors d'une requête de résolution pour un hôte mach non pleinement qualifié, on recherche d'abord mach.dom1.tp.net puis mach.dom2.tp.net.

nameserver — cette directive fournit au solveur l'adresse ip d'un serveur de noms à interroger. Si plusieurs serveurs de noms sont déclarés, le solveur commence par interroger le premier. Après un délai d'attente (5 secondes), si aucune réponse n'est parvenue, il interroge le deuxième, puis peut recommencer plusieurs cycles avec des délais différents d'attente avant d'afficher son échec.

Installation de bind

Rien de plus simple il faut utiliser *sympatic* ou **apt-get install**. La version actuelle est *bind9*.

Vous devez avoir deux paquetages logiciels :

bind9 — qui contient le démon named faisant office de serveur de noms.

dnsutils — qui contient les outils de test des serveurs de noms (host, dig, dnsquery, nslookup, ...).

Bind est maintenant installé mais non configuré.

Explication de la configuration de bind

Fichier /etc/bind/named.conf

Comme tout démon sous unix, named commence par lire un fichier de configuration. Ce fichier du répertoire /etc -- dont le format est est spécifique à Bind -- est par défaut named.conf.

Dans ce fichier nous trouvons :

- la déclaration des zones gérées
- leur association avec les fichiers constituant la ``base de données''

On dispose de deux structures :

options — qui permet de définir des options globales de bind, comme le répertoire de travail, ...

```
options {

        directory "/etc/bind";

        pid-file "/var/run/named/named.pid";

};
```

Sur cet exemple, tous les fichiers de zones sont à mettre dans le répertoire /etc/bind. C'est aussi que vous pourrez spécifier un forwarder.

zone — qui permet de définir le type et l'emplacement du fichier des données pour une zone.

Les trois types que vous allez rencontrer aujourd'hui sont les suivants :

- serveur maître primaire (primary master) d'une zone

```
zone "tp.net" in {

        type = master;

        file = "db.tp";

};
```

  Ce qui se lit ``je suis serveur maître primaire de la zone tp.net, je récupère les données de cette zone dans le fichier db.tp (du répertoire /var/named)

- serveur esclave (slave) d'une zone
- configuration de la localisation des serveurs racines

```
zone "." {
        type hint;
        file "/etc/bind/db.root";
};
```

La différence essentielle entre un serveur maître primaire et un esclave est la provenance de leurs données. Le maître primaire obtient ses données à partir de fichiers alors que l'esclave les télécharge depuis un autre serveur de nom.

Fichiers de zones

Le format des fichiers de configuration est défini de manière standard dans les RFC (1034 et 1035) et est donc le même quel que soit le serveur de DNS choisi. Ces fichiers contiennent des enregistrements de ressource (RR Ressource Records). Chaque enregistrement de ressource est un quintuplet :

Nom_de_domaine	nous donne le domaine que concerne l'enregistrement
Durée_de_vie	le champs durée de vie donne une indication sur la stabilité dans le temps de cet enregistrement (une information très stable se verra affecter une valeur comme 86400 (1 journée en seconde), une information peu stable se verra affecter une valeur comme 60 (1 minute).
Classe	la classe peut être IN pour Internet (cas par défaut), que nous utiliserons toujours, CS (CSNET), CH (Chaos), HS (Hesiod). Ces trois dernières valeurs ne sont en pratique jamais utilisées.
Type/Valeur	indique le type d'enregistrement. Les types les plus important sont les suivants :

SOA	Start Of Authority
A	correspondance nom-adresse : une adresse ip
MX	relais de messagerie
NS	serveur de nom
CNAME	nom canonique (alias) : le correspondant de cet alias
PTR	correspondance adresse-nom : un nom d'hôte

Chaque fichier de zone, à l'aide de ces enregistrement va définir en particulier :

- Qui a l'autorité ? L'autorité est détenu par LE serveur maître. Cette information est renseignée par un enregistrement SOA (un et un seul). Un enregistrement pour le serveur maître primaire.

- Quels sont les serveurs de noms de la zone : c'est le rôle des enregistrement NS. Il peut y avoir plusieurs serveurs de noms pour une zone, un maître primaire et des esclaves. Chacun va faire l'objet d'un enregistrement NS. Un enregistrement NS pour chaque serveur de nom de la zone.

- Enfin dans une zone de résolution directe, on va définir la correspondance nom-adresse à l'aide des enregistrement A ; dans une zone de résolution inverse, on va définir la correspondance adresse-nom à l'aide des enregistrement PTR ; Un enregistrement A ou PTR (selon la zone) pour chaque hôte de la zone.

Mise en place d'un serveur primaire.

Configurer /etc/bind/named.conf pour gérer la zone de recherche directe *intra*, la zone de recherche inverse *114.168.192.in-addr.arpa* et un *redirecteur* vers *192.168.114.254*.

Créer ensuite les deux fichiers de zone correspondant.

Le démon bind peut être contrôlé de deux manières :

- graphiquement avec le *régalge des services*,

- en ligne de commande avec **/etc/init.d/bind9 restart** ou **/etc/init.d/bind9/reload**.

A chaque (re)démarrage de *bind*, les fichiers de configuration sont relus.

Pour suivre l'activité du démon named utilisez la commande suivante : **tail -f /var/log/messages**.

Un démarrage réussi ressemble à ceci :

```
#/etc/init.d/bind9 restart
* Stopping domain name service... bind
* Starting domain name service... bind
```

Il vous faut tester avec la commande **dig** que votre résolution fonctionne pour *www.intra*, *user1.intra*, *user2.intra*,*user3.intra* et *192.168.114.X*.

Un très bon tutorial est disponible à l'adresse suivante : `http://doc.ubuntu-fr.org/bind9`

Mise en place des sites Web

Dans cette partie nous allons utiliser le serveur hhtp *apache2*.

Architecture Apache

Apache est le serveur web le plus utilisé avec près de 65% du marché (contre moins de 35% aux différents serveurs Web Microsoft). Parmi les facteurs du succès d'apache, on peut citer :

- le mode de distribution d'apache qui est fournit avec ses sources et permet (gratuitement) une utilisation non commerciale aussi bien que commerciale.

- l'architecture modulaire ; les utilisateurs d'apache peuvent facilement rajouter des fonctionnalités et adapter apache à leur propre besoin.

- la portabilité : il existe des version d'Apache pour tous les Unix (dont Linux bien sur), mais aussi pour windows, ...

- enfin robustesse et sécurité

Modules

Les modules peuvent être intégrés dans le programme binaire httpd au moment de la compilation. Il est également possible de compiler à part des modules en tant qu'objets dynamiques partagés (Dynamic Shared Objects : DSOs)

existant séparément du fichier binaire principal httpd. Les modules DSO peuvent être compilés en même temps que le serveur, ou après, au moyen de l'outil Apache pour les extensions (apxs). Les modules DSO peuvent être chargé dans `httpd.conf` avec la directive `LoadModule`.

Modules standards

mod_dir, mod_autoindex	Chargement automatique d'un fichier (`index.html` par exemple) et création automatique de la liste des fichiers d'un répertoire.
mod_alias	Association de différentes parties du système de fichier de l'hôte dans l'arborescence des documents, et *redirection* des URL.
mod_userdir	Permet de gérer les répertoires personnels des utilisateurs (votre `public_html`).
mod_cgi	Pour activer la passerelle CGI (Common Gateway Interface)

Modules non standards :

mod_php	Pour faire du php...
mod_dav	`mod_dav` est un module apache pour intégrer des fonctionnalité DAV (RFC 2518) à votre serveur web apache. DAV signifie: "Distributed authoring and Versioning". DAV peut être vu comme un ensemble d'extensions au protocole HTTP qui permettent d'éditer et de gérer des fichiers sur des serveur web distants. Cela permet d'éviter des mises à jour par ftp.
mod_perl	Permet d'écrire des scripts CGI sans devoir créer un processus à chaque requête et fourni une interface perl au serveur (pour rajouter des modules écrits en perl).

Dans la distribution que nous utilisons les modules se trouveront une fois apache insatallé dans `/etc/apache2/mods-available`.

Installation binaire ou source ?

Une installation d'une version *binaire* d'apache vous permet de passer outre les problèmes de compilation du logiciel. C'est donc la solution pour obtenir rapidement un serveur opérationnel. Cette méthode permet aussi d'obtenir des mises à jour automatiques mais elle présente certains inconvénients :

- Une version binaire a été compilée pour une machine et un système d'exploitation particuliers. Il n'y a pas forcément une version disponible pour vos besoins.
- Les versions compilées d'apache, spécifiquement celles disponibles sur des cdroms ou sur les mirroirs de la distribution , sont souvent plus anciennes que la dernière version source. Vous pouvez ne pas disposer du dernier correctif logiciel (ce qui peut affecter la sécurité de votre serveur) ou ne pas disposer d'une nouvelle fonctionnalité.
- Les versions binaires sont préconfigurées, ce qui peut, pour certaines, vous empêcher d'adapter apache à vos besoins particuliers (utilisation du module PHP par exemple).

  La compilation adaptable à merci possède aussi ses inconvénients, l'administrateur doit se tenir au courant des correctifs de sécurité de la hiérarchie des bibliothèques utilisées et recompiler le serveur et les différentes bibliothèques à chaque changement.

Compilation

Si vous souhaitez compiler apache, **ce que nous ne ferons pas dans ce TP** vous devrez suivre les étapes suivantes :

1. Télécharger les sources sur `http://www.apache.org`.
2. Extraire les fichiers

```
$ gzip -d httpd-2_0_NN.tar.gz
$ tar xvf httpd-2_0_NN.tar
```

3. Préparer "the source tree". Par exmple pour utiliser les modules `mod_rewrite` et `mod_speling` en utilisant plus tard le méchanisme DSO :

```
 $ CC="pgcc" CFLAGS="-O2" \
./configure --prefix=/sw/pkg/apache \
--enable-rewrite=shared \
--enable-speling=shared
```

4. La compilation :

```
$ make
```

5. L'installation :

```
$ make install
```

6. La configuration :

```
$ vi PREFIX/conf/httpd.conf
```

7. Le test :

```
$ PREFIX/bin/apachectl start
```

Paquets d'installation

Nous allons dans ce TP utiliser des paquets d'insatllation (une distriution binaire). Devant plus tard installer *php5* vous pouvez l'installer directement ce qui installera automatiquement une version d'apache.

Découverte de l'installation de apache

La premiere tache est de découvrir où est installé apache et quels sont les fichiers de configuration. Vous pouvez réaliser cette opération avec le gestionnaire de paquets graphique ou avec **apt-get**.

Démarrage/arrêt d'apache

Apache peut-être démaré manuellement avec un script de lancement (`/usr/sbin/apache2ctl`) est fourni. Ce script prend un argument en paramètre : start, stop, restart et d'autres paramètres.

Apache peut-aussi être lancé au boot via le script `/etc/init.d/apache2`, c'est cette dernière méthode que nous utiliserons. En testant vous devez obtenir quelque chose de la forme :

```
#/etc/init.d/apache2 restart
 * Forcing reload of web server (apache2)...                              [ OK ]
```

Vous pouvez aussi observer les fichiers de log du répertoire `/etc/var/log/apache2`.

Fichiers de configuration

Le fichier de configuration principal est `httpd.conf`. Sous la *Ubuntu* celui-ci est appelé via le fichier `apache2.conf`, c'est dans ce fichier que nous trouverons nos premières directives :

ServerName	est utilisée pour déterminer comment former les URLs s'auto référençant. Par exemple, quand un client requiert un répertoire, mais n'inclut pas le / final dans le nom du répertoire, Apache doit rediriger le client vers le nom complet, incluant le / final permettant ainsi au client de résoudre correctement les références relatives contenues dans le document. Vous positionnerez *ServeurName à serveur.intra.*
ServerRoot	permet de spécifier le répertoire racine à du serveur.
DocumentRoot	permet de spécifier le répertoire racine à du serveur.
ServerAdmin	définit l'adresse e-mail que le serveur inclut dans tout message d'erreur retourné au client.
User	Utilisateur utiliser par le serveur apache

Group Groupe d'appartenance d'apache.

LogLevel Le niveau de log, je vous conseil de le mettre au niveau le plus élevé pour ce TP.

Vous ne trouverez que `ServerRoot, LogLevel,Group et User` les autres sont dans l'hôte virtuelle dont vous trouverez la définition dans `/etc/apache2/sites-available/default`. Vous pourrez vous inspirer de ce dernier fichier pour créér plus tard vos hôte virtuels.

Un autre bon tutoriel est disponible à l'URL suivante : `http://doc.ubuntu-fr.org/apache2`

Sites simples

Nous allons créer les sites `http://www.intra/`, `http://user1.intra/` ,`http://user2.intra/`, `http://user3.intra/` , `http://www.S3X.ens.src/`,`http://secure.S3X.ens.src/`.

Commencer par créer les utilisateurs (user1, user2, user3) avec la commande **adduser** et affecter leurs un mot de passe (pass) avec la commande **passwd**.

Puis, recréer l'abrorescence des répertore de la partie Windows complétée par le répetoire `/web/internet/www/replisting`. Vous pouvez utiliser la commande **mkdir** et enventuellement **chmod**.

```
# tree web
web
|-- internet
|   |-- secure
|   `-- www
|       `-- replisting
|
`-- intranet
    |-- users
    |   |-- user1
    |   |-- user2
    |   `-- user3
    `-- www
```

Vérifier avec la commande **tree** que vous avez bien l'arborescence précédente.

Pour identifier nos sites, nous allons utiliser des hôtes virtuels par nom, il est aussi possible d'utiliser des hôtes virtuels par ports ou adresses IP.

Exemple 3.2. Exmple d'hôtes virtuels par noms

L'exemple suivant décrit la création des hôtes virtuels : `www.domain.tld` et `www.otherdomain.tld`.

```
NameVirtualHost *:80

<VirtualHost *:80>
ServerName www.domain.tld
ServerAlias domain.tld *.domain.tld
DocumentRoot /www/domain
</VirtualHost>

<VirtualHost *:80>
ServerName www.otherdomain.tld
DocumentRoot /www/otherdomain
</VirtualHost>
```

Créer vos hôtes virtuels dans puis placer dans chaque répertoire un fichier permettant de l'identifier et enfin tester avec votre navigateur. Les sites doivent être créés dans `sites-avalaible` et un lien symbolique doit être créé dans `sites-enabled`.

Astuce

Vous pouvez placer la définition de vos hôtes virtuels dans le même fichier, ce qui vous facilitera la tâche.

Personnalisation de `www.intra` et de `www.S3X.ens.src`

Fichiers par défaut et listing des répertoires

Nous souhaitons pour nos deux sites que les fichiers `index.htm` puis `index.html` soient servis en premier et que le listing des répertoires soit impossible sur `http://www.S3X.ens.src` mais soit possible pour `http://www.S3X.ens.src/replisting`. Dans un soucis de tester vous ne devez pas hésiter à renommer vos fichiers.

Pour configurer *apache* vous aurez besoins des directives :

- DirectoryIndex (du module mod_dir)
- <Directory>
- Options

Les directives ont une portée. Les directives placées dans le fichier de configuration s'applique à l'ensemble du serveur. Pour modifier la configuration pour une partie seulement du serveur il faut mettre les directives correspondantes dans des sections :

<Directory>, <DirectoryMatch>	gestion d'un répertoire (chemin local)
<Files>, <FilesMatch>	gestion d'un fichier (chemin local)
<Location>, <LocationMatch>	gestion d'une URL (chemin depuis la racine du serveur)

Pour chaque localisation, il est possible de spécifier de fonctionnalité avec la directive `Options`. La directive Options contrôle quelles fonctions du serveur sont disponibles dans un répertoire particulier ou un serveur virtuel. Dans les options, on trouve :

ExecCGI	Le serveur permet d'exécution de scripts CGI
FollowSymLinks	Le serveur est autorisé à suivre les liens symboliques dans ce répertoire.
Includes	Les SSI (Server Side Include) sont autorisés, sauf les commandes #exec et #include des scripts CGI.
IncludesNOEXEC	des scripts CGI.
Indexes	autorise le listage du contenu d'un répertoire.

Il est aussi possible de filter les clients en fonction de leurs adresses IP. Les directives `Allow` (accorder des droits) et `Deny` (en enlever) doivent être utilisées. De plus la directive `Order` permet de spécifier dans quel ordre on applique les droits.

Répertoire de publication utiulisateur (public_html)

Vous avez pu constaté que vous disposer d'un `public_html` ce repertoire permet d'acceder au répertoire `public_html` du *home directory* via l'url `http://www.intra/~nom_user`. Pour avoir les *home directory* vous devez commencer par créer les trois utilisateurs (**adduser**) système *user1*, *user2* et *user3*, tous les trois auront comme mot de passe *pass*. Une fois les utilsateurs créés vous devez créer dans leur *home directory* un répertoire `public_html` dont ils seront les propriétaires respectifs, *apache* doit pouvoir lire le contenu du répertoire et pouvoir y entrer. Les commandes **su** ou **chown** et **chmod** peuvent vous aider.

Nous allons mettre en oeuvre des répertoires de publication des utilisateurs, pour www.intra, elle devra donc être interdite pour les autres sites. Pour permettre l'utilisation de répertoire de publication utilisateur, il faut pour ce faire utiliser le module `mod_user_dir` et la directive `UserDir`.

Sous la ubuntu le fichier de configuration du module est dans `/etc/apache2/mods-available/`, il se nomme `userdir.conf`. Cependant, ce module n'est pas disponible, il faut pour qu'il le soit qu'un lien symbolique (**ln -s**) soit créé dans `/etc/apache2/mods-enabled/`vers `userdir.load`.

La solution du public dans `www.intra/~userX` peut être amélioré en utilisant le répertoire de base de `userX.intra`. Pour ce faire, vous pouvez modifier la directive **UserDir**.

Secure Socket Layer (SSL)

Nous allons securiser le site secure.S3X.pedado.src en utilisant SSL.

Sous linux, il est possible de génerer un certificat serveur autosigné en utilisant *openssl*.

La ligne de commande suivante vous permet d'obtenir le sertificat :

```
#sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out
/etc/apache2/secure.crt -keyout /etc/apache2/secure.key
```

x509 -nodes	donne le type de certificats souhaité
days	spécifie le nombre de jours
newkey	spécifier le type de clef
out	donne le chemin de génération du certificat
keyout	donne le chemin de la clef privée

Une fois le certificat obtenu, il nous faut configurer apache :

1. dans `/etc/apache2/ports.conf` il faut utiliser la directive Listen pour écouter sur le port 443,
2. il nous faut modifier l'hôte virtuel `secure.S3X.ens.src` qui écoute sur le port 80 pour qu'il ne réponde plus (**Deny**) et en créer un autre qui écoute sur le port 443. Dans cet hôte virtuel les directives suivantes doivent être utilisée:
   - SSLEngine
   - SSLCertificateFile
   - SSLCertificateKeyFile.
3. Pour finir, il faut recharger la configuration d'apache et tester avec un navigateur.

Une fois le site sécurisé en place, désacitiver `http://secure.S3X.ens.src`.

File Transfert Protocol (FTP)

Nous allons utiliser le serveur FTP proftpd. Nous souhaitons que *userX* puissent déposer des fichiers sur leurs sites.Il faut donc que les répertoires `/web/intranet/users/userX` soient avec des droits d'écriture (**chmod**) pour les utilisateurs concernés (**chown**).

Le serveur ftp et *proftpd*, il dispose si vous le souhaitez d'une interface graphique de configuration *gproftpd*. Lors de l'instalation vous choisirez *standalone*. Cette oprion signifie que le serveur n'est pas lancé par un *super daemon* mais qu'il est lancé au démarage ou manuellement.

Comme clients graphiques vous pourrez utiliser *nautilus ou* filezilla.

Après installation, dans un premier temps, nous allons emprisonner les utilisateurs dans leur home directory *pour cela il vous faut configurer votre serveur ftp.*

*Dans le fichier de configuration, vous devrez modifier le paramètre suivant : DefaultRoo*t (la lecture de la documentation peut vous aider).

E même vous devrez ajouter à la fin du fichier la directive suivante pour autoriser les login :

```
<Limit Login>
   AllowAll
</Lilit>
```

Les utilisateurs étant vérouillé dans leur *home directory* comment faire pour qu'is puissent accèder à leur répertoire de publication ? La commande suivante devrait pouvoir vous aider :

```
sudo mount --bind dossier_source dossier_ftp
```

La commande mount sera perdu au redémarrage, nous allons donc créé un script qui sera exécuté à chaque démarrage :

1. Créer dans `/etc/init.d` un fichier mes_montages qui contient les commandes **mount**;
2. Mettre le fichier mes_montages en exécution (**chmod**);
3. Modifier les scripts de démarrage avec la commande

   ```
   update-rc.d mes_montages defaults
   ```

WebDAV (Web-based Distributed Authoring and Versioning)

Nous allons permettre un accès WebDav pour le site `http://www.intra`. Pour `http://www.intra`, nous allons autoriser le download pour let l'upload pour l'utilisateuyr apache *admin*. *admin* est un utilisateur *apache*, ce n'est pas un utilisateur système, *apache* peut maintenir sa propre base de compte.

Apache a besoin deux modules obligatoire et un optionnel pour WebDav :

mod_dav	c'est le module principale qui permet à apache de gérer les extensions WebDAV,
mod_dav_fs	c'est un module support de `mod_dav` qui permet d'interagir avec le système de fichier.
mod_dav_lock	est un module non indispensable ici qui permet de gérer les verrous.

Etablir les liens nécessaires.

Le module `mod_dav` nous fournit la directive `Dav` qui permet d'activer WebDAV pour une localisation donnée.

Le module `mod_dav_fs` nous fourni la directive `DavLockDB` qui spécifie ou se trouve le repertoire utilisé pour gérer les verrous.
Cette directive est postionnée sous ubuntu dans dav_fs.conf.

Ajouter la directive `DAV` au site `http://www.intra`.

Relancer apache et tester avec nautilus en utilisant l'URL suivante `dav://www.intra`. Vous devez avoir un accès en lecture seule.

Le client WebDAV étant un peu facécieu vous pouvez utiliser le système de fichier DAVFS :

1. Installer le module `davfs2`;
2. Créer un répertoire, *le point de* montage (par exemple `/mnt/WebDav`);
3. *Monter* le volume avec la commande **mount -t davfs url point_de_montage**;
4. Pour démonter le volume, après usage, vous devez utiliser **umount**.

Apache supporte de nombreux modules d'authentification, nous allons utiliser le plus simple, l'authentification basique (`mod_auth_basic`) qui fait circuler en clair le mot de passe. Vérifier si le module est installé puis suivre la procèdure suivante :

1. Création d'un fichier contenant les mots de passe et donc celui de admin :

```
htpasswd -c /etc/apache2/passwd.dav admin
chown root:www-data /etc/apache2/passwd.dav
chmod 640 /etc/apache2/passwd.dav
```

Seuls les membres du groupe d'apache (www-data) et root(root) pourront accèder à ce fichier.

2. Ajouter les directives suivantes à votre site :

```
<Directory /web/intranet/www>
 DAV On
 AuthType Basic
 AuthName "BoxBox"
 AuthUserFile /etc/apache2/passwd.dav

  <Limit PUT POST DELETE PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
         Require user admin
  </Limit>
</Directory>
```

A ce stade un mot de passe doit être demandé pour *`dav://admin@www.intra` mais pas pour dav://www.intra, la limitation n'étant pas appliquée à `GET`. Cepandant admin ne peut toujours pas* uploader.

3. Attribution des droits d'écrire pour apache sur `/web/intranet/www` :

```
chown -R www-data:www-data /web/intranet/www
chmod -R 750 /web/intranet/www
```

4. Tester avec un client WebDav et un client http.

.htaccess

Le fichier `.htaccess` (distributed configuration files) permet de redéfinir les configuration des répertoire. Un `.htaccess` vaut pour le répertoire qui le contient et pour les répertoires englobés.

L'utilisation d'un `.htaccess` est permise en utilisant les directive : `AccessFileName`, `AllowOverride`.

Sur le site `http://user1.intra` autoriser toute les redéfinitions de `AuthConfig` et `FileInfo`, puis essayer de réaliser les opérations suivantes :

1. Mettre en place des pages d'erreurs personnalisée en utilisant la directive `ErrorDocument`.

2. Protéger par mot de passe un répertoire pour un amis (*MonAmis*).

Montrer votre solution à un enseignant pour qu'il en vérifie la sécurité.

nstallation du Système de gestion de base de données MySQL

Nous allons intaller le SGBD (Système de gestion de base de données) *MySQL* :

1. Nous allons commencer avec installer la version la plus recente :

```
sudo apt-get install mysql-server
```

2. Lancer le SGBD :

```
sudo /etc/init.d/mysql restart
```

3. Définir le mot de passe de l'administrateur du SGBD :

```
sudo mysqladmin -u root password mot_de_passe
```

4. Tester avec un client :

```
mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 5.0.38-Ubuntu_0ubuntu1.1-log Ubuntu 7.04 distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
+--------------------+
2 rows in set (0.00 sec)
```

Nous allons maintenant un compte (utilisateur) sur le SGBD : *joomla*. L'utilisateur *joomla* devra posséder une base de donnée du même nom sur laquelle il aura tous les privilèges. Vous aurez besoin des commandes suivantes :

- **create user** Création d'un utilisateur,
- **create database**Création d'une base,
- **grant all on nom_base.* to nom_utilisateur@localhost;**attribution de tous les droits sur une base pour un utilisateur précis.
- **SET PASSWORD FOR 'toto'@'%' = PASSWORD('pass');**Définie le mot de passe pass pour l'utilisateur toto.

Pour finir le fichier de configuration est `/etc/mysql/my.cnf` dans ce fichier vous contaterez que seules les connexions sur 127.0.0.1 sont autorisées (`bins-address`) mais cela ne va pas nous géner.

Installation de Hypertext Preprocessor (PHP)

Vérifier si PHP est installer, sinon l'installer.

Vous trouverez le fichier de configuration ici : `/etc/php5/apache2/php.ini` .

Déploiement d'applications web

Nous allons installer *phpmyadmin* et *joomla.*

phpmyadmin sera déployé sur `http://www.intra` et *joomla* sur chaqu'un des comptes utilisateur.

Installer phpymyadmin, utiliser la directive *alias* dans la configuration de `http://www.intra` et créer dans `/web/intranet/www`un répertoire vide de même nom que l'alias.

Astuce

Si vous obtenez une erreur 500, penser à regarder les logs ils peuvent être instructifs.

Tester phpmyadmin en *root* et en *joomla*avec pour ce dernier une création de table.

Astuce

Si vous ne pouvez vous connecter la solution est peut-etre dans `/etc/phpmyadmin/config.inc.php`

Pour *joomla* il faudra copier les fichiers d'installation sur chaque répertoire et suivre la procèdure. Sous Linux, il est pour habitude d'installer les fichiers non propre à la distribution dans `/opt`. C'est donc là que vous créerez votre répetoire contenant les sources de *joomla*.

Redirection et réécriture d'URL

Apache peut réécrire à la volée les URL pour cela il vous faut aciver le `mod_rewrite`.

Les directive a utiliser son `RewriteEngine`, `RewriteRule`, enventuellement `RewriteCond` et `RewriteOptions`.

Exemple 3.3. Réécriture triviale

La régle `RewriteRule ^truc-client$ fichier.html [L]`est comprise comme suit :

- ^ le debut de l'expression

  $ la fin de l'expression

  truc-client l'expression a réécrire

- fichier.html le resultat de la réécriture

- [L] Un drapeau) signifiant que cette règle est la dernière à appliquer dans ce cas.

Il faut remarquer que cet exemple est pédagogique, nous pouvions faire la même chose avec la directive `Redirect`.

Mettre en oeuvre cet exemple sur `http://www.S3X.pedao.src/pas-la` qui doit être réécrite vers un fichier présent sur le site.

Les règles de réécriture sont complexes voici quels expressions régulières utiles et quelques drapeaux possibles.

Tableau 3.1. Expression régulières

expression	**signification**
.	n'importe quel caractère
[azer]	n'importe lequel de cette liste de caractères
blanc|noir	alternative, soit « blanc », soit « noir »
+	Une ou N occurrence(s) de l'expression qui précède (N >= 1)
*	Zéro ou N occurrence(s) de l'expression qui précède (N>=0)
(texte)	Groupement permettant l'utilisation des références inverses ($1,... $n) Est aussi utilisé pour délimiter une alternative comme dans (blanc|noir)
^	ancre de début de ligne
$	ancre de fin de ligne
\	permet d'échapper tout caractère qui suit et lui ôter sa signification particulière, par exemple \.

Samba

Le but de ce tp est de partager des ressources (fichiers) sur un réseau hétérogène (composé de machines sous Window et Linux). Ce partage de ressources se fait en implantant sous linux les protocoles réseaux (pour les couches supérieures à TCP/IP) utilisés par les systèmes d'exploitation microsoft.

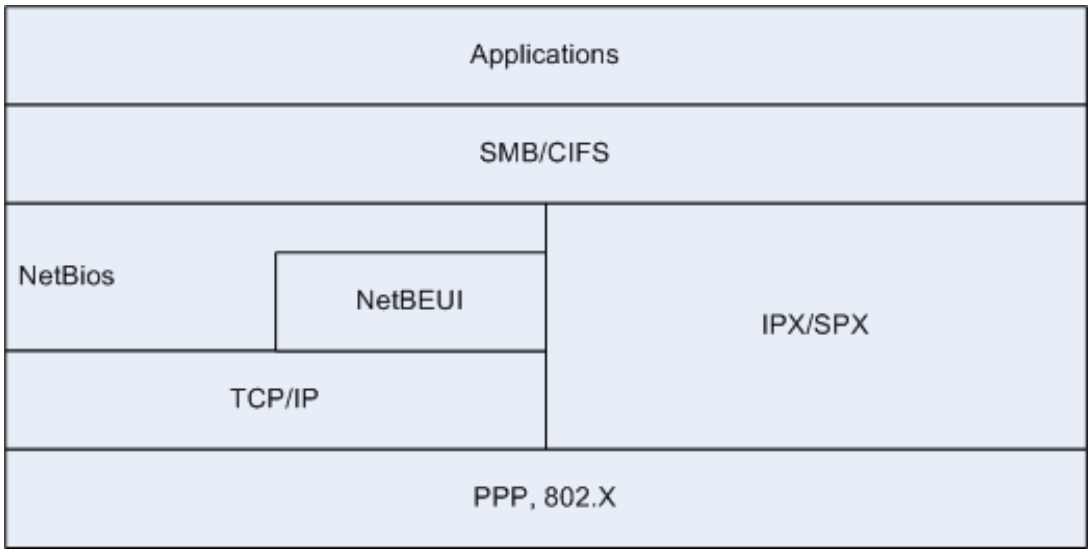
Nous utiliserons donc deux machines, l'une sous linux et l'autre sous Windows.

Vous commencerez par renommer la machine Windows en machine-X ou X est le numéro de votre disque dur.

Réseaux SMB/CIFS

SMB (Server Message Block) est un protocole pour le partage des imprimantes et de fichiers. SMB est un protocole client/serveur avec requête du client et réponse du serveur. Il s'appuie sur des protocoles réseaux de plus bas niveau comme NetBEUI(Network Basic Input/Output System Extended User Interface), TCP/IP (Transmission Control Protocol/Internet Protocol) ou IPX (Internetwork Packet Exchange). NetBEUI ou TCP/IP sont utilisés par l'api NetBios.

Figure 3.1. Différents protocoles mis en oeuvre



CIFS (Common Internet File System) n'est rien d'autre que le nom des nouvelles version de SMB. CIFS permet un accès à des ressources distantes sur Internet et est compatible avec les protocoles utilisés sur un réseau local. CIFS utilise TCP/IP et le DNS. La résolution de nom DNS n'est pas la seule posible mais elle est conseillée car ne reposant pas sur la diffusion donc plus rapide.

Un réseau composé de machines utilisant un système d'exploitation Microsoft utilise le protocole SMB/CIFS. Un Workgroups ou groupe de travail est un ensemble d'ordinateurs qui résident sur le même sous-réseau et qui appartiennent au même groupe SMB. Chaque ordinateur gère lui-même les ressources et les permissions d'accès à ces ressources. Un domaine est un Workgroups avec une fonctionnalité supplémentaire de contrôleur de domaine. Le contrôleur de domaine centralise la politique de sécurité en stockant une liste de couples login/password valides pour ce domaine.

Historique

En 1984, IBM développe une ``extension'' réseaux des APIs du BIOS : NetBIOS (Network Basic Input/Output System). NetBIOS offre des services aussi bien en mode connecté (session) qu'en mode non connecté (envoi de datagramme) à un groupe de PC partageant un média de diffusion commun. Ces services permettent aux applications NetBIOS de localiser une ressource, établir une connexion, envoyer et recevoir des données avec une application distante, terminer une connexion, ... Cette API nécessitait cependant en complément, un protocole réseau de bas niveau pour transférer les requêtes d'un ordinateur à l'autre. IBM à proposé un tel protocole fin 1985 : NetBEUI (NetBIOS Extended User Interface). NetBEUI a été conçu pour de petits réseaux (moins de 255 noeuds) où une machine est identifié par un nom (de 15 caractères au plus). Ce protocole est devenu très populaire en particulier chez Microsoft. L'importance grandissante d'Internet et de ses protocoles réseaux (TCP et UDP au dessus de IP) a donné lieu à des tentatives d'implantation de NetBIOS utilisant TCP/IP. Cette standardisation est décrite dans les rfc 1001 et 1002 et est connue sous le nom de NBT ou NetBIOS over (au dessus de) TCP/IP. NetBIOS est composé essentiellement de :

un service de nom	pour faire la correspondance entre un nom NetBIOS et une adresse IP
deux services de communications des données	1. datagrames

2. sessions

Fonctionnement de NetBios

Dans un réseau NetBIOS, une machine arrivant sur le réseau cherche à enregistrer son nom. Cette procédure peut se faire avec un serveur de nom ou sans serveur de nom, par une diffusion sur le réseau.

Figure 3.2. Enregistrement d'une machine sans serveur de nom (NetBios)

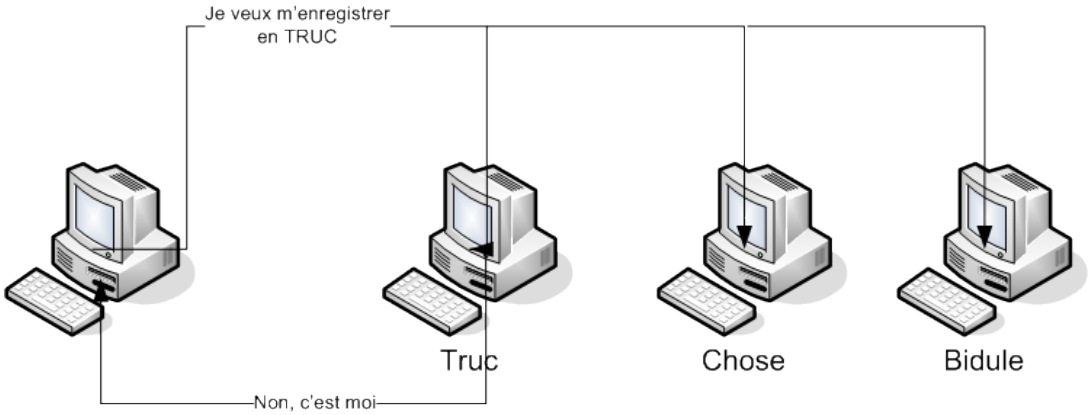
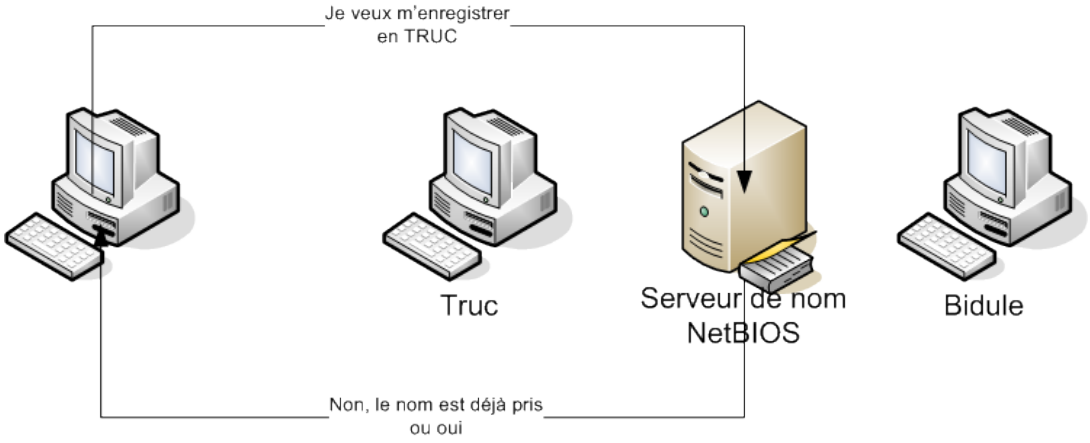


Figure 3.3. Enregistrement d'une machine avec seveur de nom (NetBios)



De même, le processus de résolution de nom peut se faire avec ou sans serveur de nom.

Figure 3.4. Résolution de nom sans seveur de nom (NetBios)

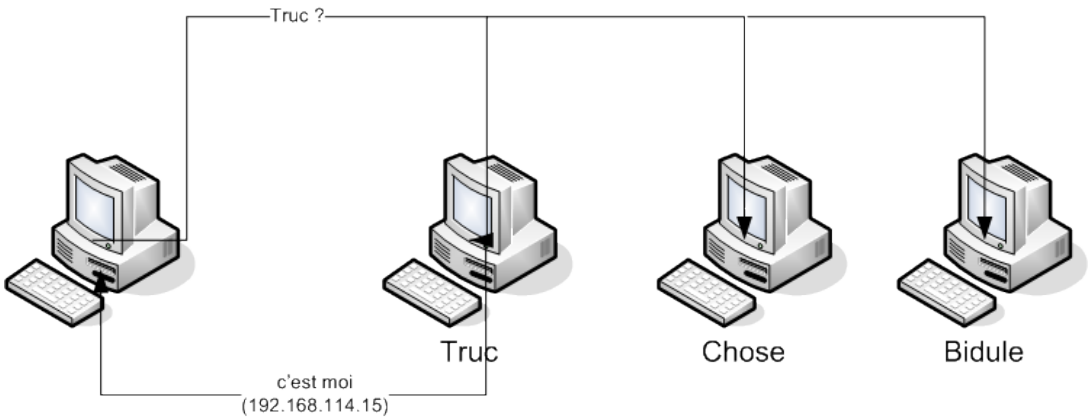
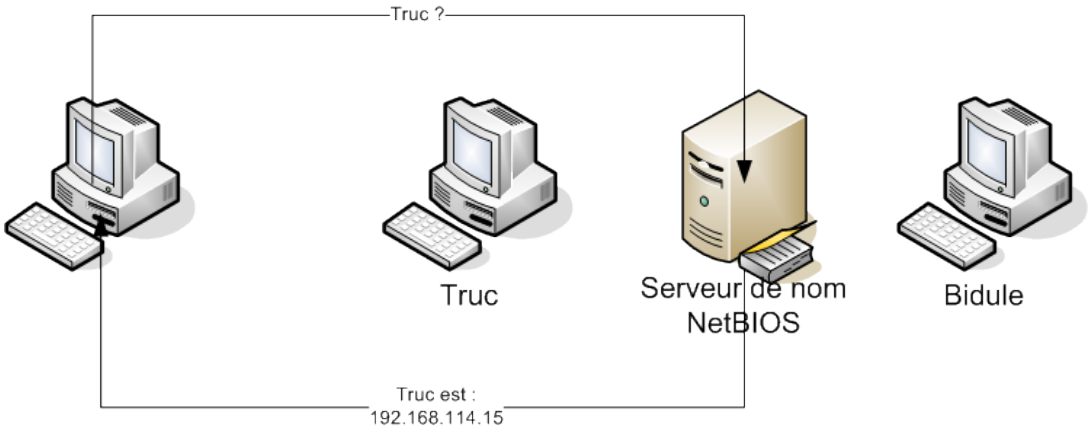


Figure 3.5. Résolution de nom avec seveur de nom (NetBios)



En plus de son nom, une machine annonce le type de service qu'elle offre. Le type de service est codé dans le 16 ième caractère du nom de la machine.

Figure 3.6. Nom NetBIOS

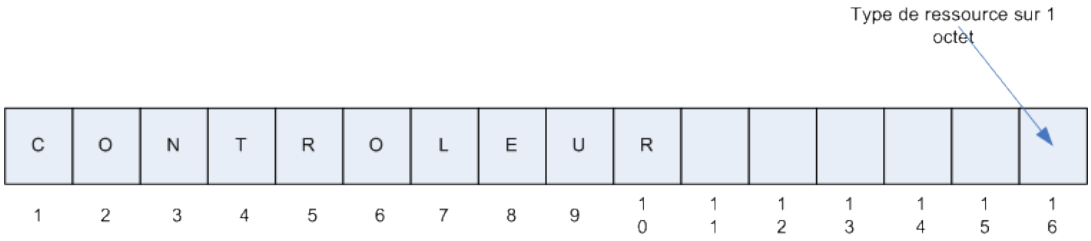


Tableau 3.2. Exemple de ressources NetBIOS

nom	valeur hexadécimale
Station de travail	00
Serveur de fichiers	20

Une fois la machine enregistrée dans le réseau, elle peut accéder à des ressources (rôle de client) ou partager des ressources (rôle de serveur). L'accès à une ressource se fait en utilisant un service :

- Datagrame
- Session

Datagrame

Le service datagrames ne propose pas de connexion stable entre deux machines. Les paquets de données sont soit :

- envoyés à une machine,
- diffusés sur l'ensemble du réseau.

Il n'y a pas de vérification du bon acheminement des paquets. C'est le protocole UDP dans un réseau NBT qui est utilisé pour les envois. Ce service datagrame est utilisé pour la résolution de nom NetBIOS en adresse IP. Les différentes primitives du service datagrame sont récapitulées dans le tableau suivant.

Tableau 3.3. Primitives du service datagrame

Envoie d'un datagramme
Diffusion d'un datagramme
Réception d'un datagramme
Réception d'un datagramme diffusé

Session

Le service session est plus complexe. Une connexion permanente à double sens est établie entre deux machines. Si des transferts se perdent, le protocole de session TCP se charge de la retransmission. Ce service session est utilisé pour l'accès à une ressource.

Les primitives du service session sont récapitulées dans le tableau suivant :

Tableau 3.4. Primitives du service session

Appeler
Ecouter
Raccrocher
Envoyer
Recevoir
Status de la session

Samba

Qu'est ce que samba

Samba est une suite d'applications *Unix* qui implante le protocole SMB. L'écriture de Samba a été initiée par . Samba permet à une machine *Unix* de communiquer avec des machines *Microsoft* (MS-DOS, Windows 3.11, Windows 95 et 98, Windows NT, Windows XP, ...) en utilisant leur protocole réseau. Samba peut transformer une machine *Unix* en serveur (sur un réseau microsoft) offrant les fonctionnalités suivantes : partage de répertoires partage d'imprimantes participation à la résolution de nom WINS (Windows Internet Name Service) participation à l'authentification de client dans un domaine L'ensemble de ces fonctionnalités est assuré par trois démons :

smbd serveur de fichiers et serveur d'imprimantes

nmbd service de nom et gestion de l'exploration

winbind pour joindre un domaine Windows que nous n'utiliserons pas dans ce TP.

Les fichiers de configuration se trouverons dans `/etc/samba/smb.conf`. Samba comprend aussi un client SMB pour accéder à un serveur SMB.

Instalation et configuration

Nous allons utiliser les paquets de configuration. Normalement vous devez avoir le paquet `smb-common` qui contient le client. Pour avoir les serveurs samba (*smbd* et *nmbd*), il vous faut installer le paquet `samba`.

Pour relancer les dameon vous pouvez utiliser **/etc/init.d/samba restart**.

Pour la configuration vous pourrez utiliser :

\etc\samba\smb.conf Le fichier principal de configuration

Des interfaces graphiques

swat	Une interface Web de configuration fournie avec *samba* que nous n'utiliserons pas car nécessitant un mot de passe root ce n'est pas dans la philosophie *ubuntu*
webmin	Une autre interface web que nous n'utiliserons pas
gsamba	Un interface graphique que vous pouvez installer. *Le menu est bugger vous pouvez la lancer en ligne de commande (gsambad).*

Test

Pour vérifier que les daemons sont bien lancés vous pouvez tester avec un client : **smbclient -U% -L 127.0.0.1**. Vous devez avoir quelque chose comme :

```
$ smbclient -U% -L 127.0.0.1
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.28a]

 Sharename       Type      Comment
 ---------       ----      -------
 print$          Disk      Printer Drivers
 IPC$            IPC       IPC Service (PC-JUB server (Samba, Ubuntu))
 PDF             Printer   PDF
Domain=[WORKGROUP] OS=[Unix] Server=[Samba 3.0.28a]

 Server               Comment
 ---------            -------
 PC-JUB               PC-JUB server (Samba, Ubuntu)

 Workgroup            Master
 ---------            -------
 WORKGROUP
```

De même vous vouvez vous connecter sur le serveur du département : **smbclient -U toto 192.168.114.254**.

Configuration: `smb.conf`

Avant de voir le fichier de configuration voici trois commandes parmis les nombreuses commandes de samba qui peuvent vous être utiles :

smbstatus permet de visualiser les connexions netBios/CIFS.

smbpasswd permet de gérer les logins et les mots de passe

smbclient permet de gérer la connexion a un partage netBios.

Le fichier de configuration est `/etc/samba/smb.conf` ce fichier est découposé en sections ([global], [homes], [netlogon], [profiles], [printers], [print$], ...). Plutôt que de le modifier directement à la main vous pouvez vous aider en installant le paquet *system-config-samba*. Ce paquet vous ajoutera dans le menu système, administration, samba un utilitaire de configuration graphique.

Partage d'un lecteur Windows avec des machines Unix

La première approche consiste à utiliser depuis linux des ressources partagées par une machine Windows.

Configuration de Windows

Sous Windows vérifier que le "Client pour réseaux Microsft" est installé. Puis, créer un partage nommé partage-x avec un contrôle total pour user1. Vérifier depuis Windows la visibilité du répertoire en utilisant netBios.

Liste des ressources depuis Unix

En utilisant smbclient avec le bon client lister les ressources netbios offerte par la machine de votre voisin.

Accès depuis Unix

Pour accèder à la ressource utilisant dans nautilus l'url `smb://@IP/Partage`.

Partage d'un répertoire Unix avec des machines Windows

Cette fois ci nous allons partager un répertoire sous Linux qui sera visible depuis Windows.

Non protégé

Créer `/home/user/partage-x` un répertoire, puis en passant par l'outil de configuration graphique permettez un partage pour tout le monde.

Il vous faut aussi changer le nom du serveur pour que que partage puisse fonctionner correctement.

Pour finir, vérifier dans `smb.conf`, l'appartion de'une nouvelle section et son contenu.

Protégé

Samba peut utiliser sa propre base de compte, nous allons utiliser cette fonction pour permttre un accès en écriture à l'utilisateur sambsa s-user sur le répertoire `/home/user/partage-f-x`.

Vous devez en premier, en utilisant l'interface d'administrateur créer l'utilisateur s-user mappé sur l'utilisateur Linux user avec comme mot de passe pass.

Un fois l'utilisateur créé, vous pouvez créer votre partage et le tester depuis une machine Windows.

Utilisation d'une machine Unix comme controleur de domaine (NT4)

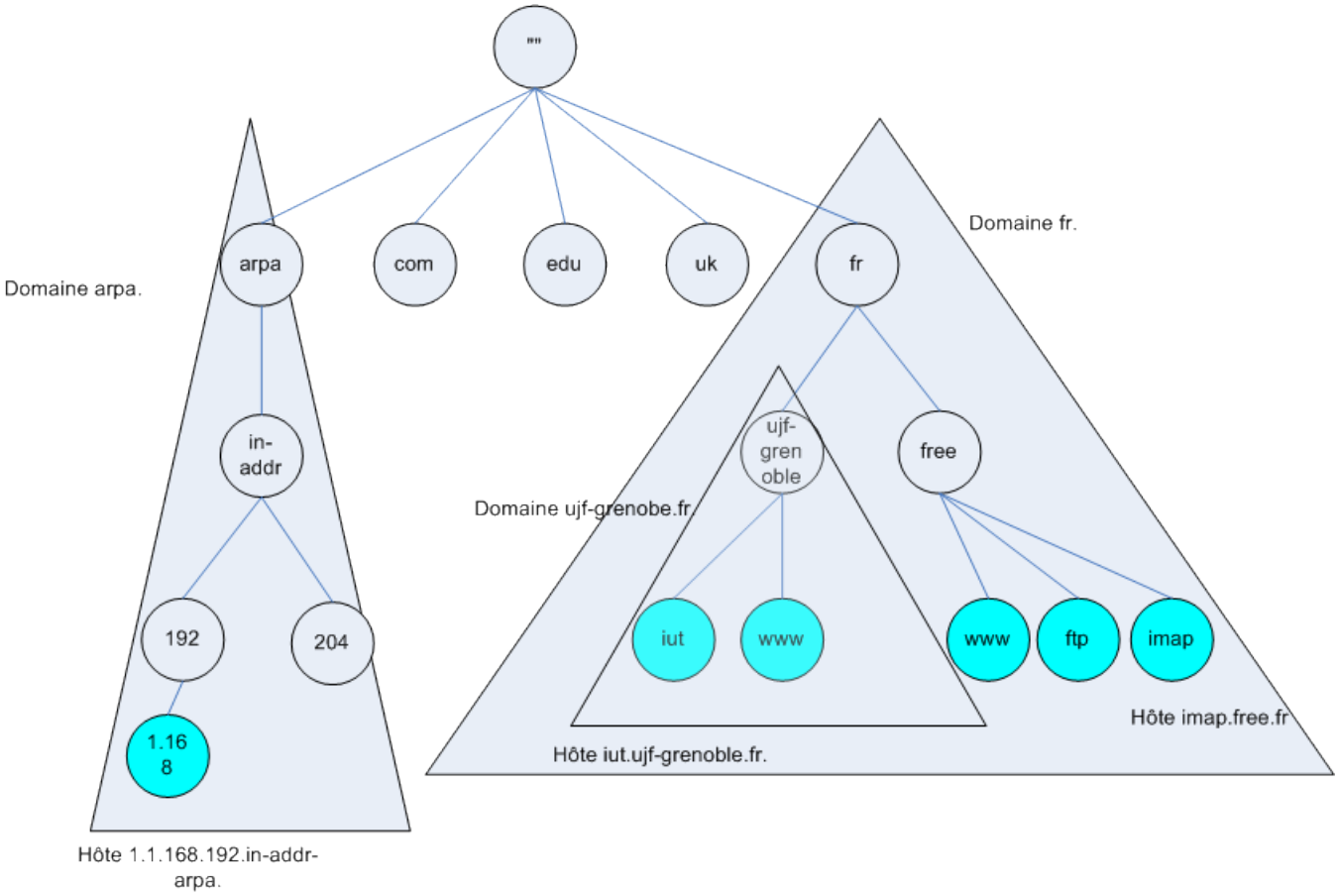
Partage des *home directory*

Glossaire

Domain Name System (DNS)

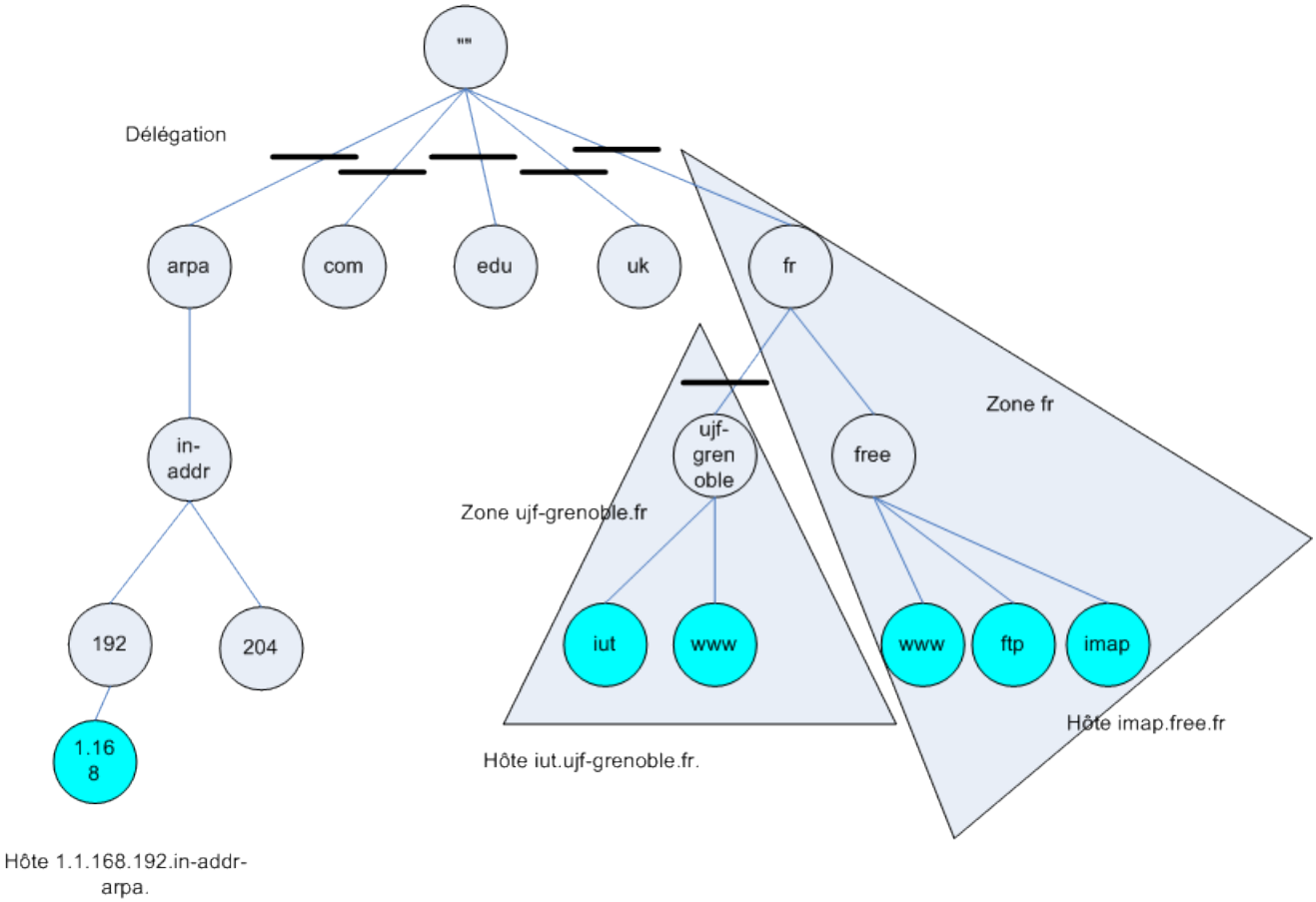
Domaine — Un noeud de l'arbre du DNS et l'ensemble de la sous-arborescence.

Figure 13. Domaine DNS



Zone — Une zone est un ensemble contigü de domaines, placés sous une même administration.

Figure 14. Zone DNS



Zone de recherche directe

Cette zone fait correspondre à un nom d'hôte une adresse IP.

Exemple 6. Zone de recherche directe toto.fr.

```
@ IN SOA pc3.toto.fr. administrateur.pc3.toto.fr.
   ( 2 ; serial number
   3600 ; refresh
   600 ; retry
   86400 ; expire
   3600 ) ; minimum TTL

@ IN NS pc3.toto.fr.

pc1 IN    A    200.50.12.1
pc2 IN    A    200.50.12.2
pc3 IN    A    200.50.12.3
```

Zone de recherche inverse

Cette zone fait correspondre un adresse IP à un nom d'hôte.

Exemple 7. Zone de recherche inverse 12.50.200.in-addr.arpa.

```
@ IN SOA pc3.toto.fr. administrateur.pc3.toto.fr.
   ( 2 ; serial number
   3600 ; refresh
   600 ; retry
   86400 ; expire
   3600 ) ; minimum TTL
@ IN NS pc3.toto.fr.
1 PTR pc1.toto.fr.
2 IN PTR pc2.toto.fr.
3 IN PTR pc3.toto.fr.
```

Serveur primaire	Un serveur primaire est un serveur qui fait autorité pour une zone, il existe un et un seul serveur primaire par domaine.
Serveur secondaire	Un serveur secondaire est un serveur qui obtient au moins un fichier de zone q'un autre serveur r de nom qui détient l'autorité pour la zone considérée.
Transfert de zone	Action qui permet à un serveur secondaire de récupérer via le réseau, au prés d'un serveur primaire un fichier de zone.
FQDN	Fully Qualified Domain Name, soit Nom de Domaine Totalement Qualifié, nom de domaine qui part de la racine de l'espace de nommage jusqu'au noeud recherché.
Type	Un même nom-domaine peut avoir des informations de plusieurs types : une adresse IPv4 (type A), une adresse IPv6 (type AAAA), un nom réel si le nom-domaine est en fait un surnom (type CNAME), un relais de courrier (type MX), etc. Dans chaque zone un certain nombre d'enregistrements sont obligatoires. Il s'agit de: * Un enregistrement Start Of Aauthority (SOA). Cet enregistrement décrit la zone. Des enregistrements Name Serveur (NS). Ces enregistrements décrivent la liste des serveurs de noms pour la zone.
Délégation	Une délégation permet d'autoriser un autre serveur DNS à contrôler une partie des enregistrements de la zone. La délégation est totale.
DNS Round-robin	Dans le Système DNS un même nom peut avoir plusieurs adresse, ce qui permet si les adresses sont utilisée les unes à la suite des autres de répartir la charge.

Exemple 8. Les serveurs DNS racine

```
;       This file holds the information on root name servers needed to
;       initialize cache of Internet domain name servers
;       (e.g. reference this file in the "cache  .  <file>"
;       configuration file of BIND domain name servers).
;
;       This file is made available by InterNIC
;       under anonymous FTP as
;           file                /domain/named.root
;           on server           FTP.INTERNIC.NET
;       -OR-                    RS.INTERNIC.NET
;
;       last update:    Jan 29, 2004
;       related version of root zone:   2004012900
;
;
; formerly NS.INTERNIC.NET
;
.                        3600000  IN  NS    A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.      3600000      A     198.41.0.4
;
; formerly NS1.ISI.EDU
;
.                        3600000      NS    B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.      3600000      A     192.228.79.201
;
; formerly C.PSI.NET
;
.                        3600000      NS    C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET.      3600000      A     192.33.4.12
;
; formerly TERP.UMD.EDU
;
.                        3600000      NS    D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET.      3600000      A     128.8.10.90
;
; formerly NS.NASA.GOV
;
.                        3600000      NS    E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET.      3600000      A     192.203.230.10
;
;...
; End of File
```

Requête récursive — Le serveur de noms contacté prend en charge la totalité de la requête et renvoie la réponse.

Requête itérative — Le serveur de noms contacté répond en donnant le nom du serveur à contacter ou la réponse si il la possède.

Vocabulaire Windows

Internet Server Application Programming Interface (ISAPI) — Méthode standard dédiée à l'écriture des programmes communiquant avec les serveurs Web par le biais d'OLE.

Object Linking and Embedding (OLE) — Technique mise au point par Microsoft pour faciliter l'échange de données entre les programmes sous Windows.

Microsoft Management Console (MMC) — Microsoft Management Console (MMC) contient et affiche les outils d'administration créés par Microsoft et d'autres éditeurs de logiciels. Ces outils s'appellent des composants logiciels enfichables, et ils permettent de gérer les composants logiciels, matériels et réseau de Windows. Plusieurs outils

	du dossier Outils d’administration, tels que Gestion de l’ordinateur, sont des composants logiciels enfichables MMC.
Microsoft management console Snap In Control file	
CPL	
cpl	