DOCUMENTATION TECHNIQUE : Installer des applications sous Linux

Description de la ressource

Propriétés	Description
Intitulé long	Fiches techniques de manipulation d'outils sous Linux
Formation concernée	BTS SIO 1ère année
Matière	MTI
Présentation	Vous trouverez dans cette documentation de l'assistance pour manipuler les outils suivants : • Généralités sur Linux (Version Ubuntu basée sur Debian) • Serveur DNS • Serveur Web Apache • SGBD MySQL • Serveur FTP ProFTPd • Cryptage SSL/TLS avec Open SSL Chaque fiche (hormis Linux) propose • une ***procédure*** à appliquer pour la mise en place de l'outil technique, ainsi qu'un bref descriptif des principaux fichiers de configuration à manipuler. Cette procédure décrit les étapes à respecter pour installer, paramétrer et mettre en activité les services. Elle ne prend pas en compte les spécificités d'un contexte et ne fournit pas d'explications techniques • une partie plus explicative sur les directives, fichiers de configuration et ***mode opératoire***.
Mots-clés	DNS, DHCP, FTP, HTTP, SSL/TLS Bind, ProFTP, MySQL, Apache
Version	v 1.2
Date de publication	Septembre 2012

Contenu

FICHE 1 : LINUX, QUELQUES GENERALITES

PRESENTATION

L'utilisation de l'environnement Linux passe par la maîtrise des éléments suivants :

- l'utilisation du *shell*
- l'arborescence des fichiers du SGF (/etc, /var, …)
- la configuration des options réseau (IP, DHCP, DNS, Routage/Passerelle)
- l'installation et la gestion du fonctionnement des services (apt)
- l'utilisation et le paramétrage de fichiers de configuration

Principes
Avant toute installation, vous devez :

- mettre à jour les paquetages lors de la première utilisation de l'OS : *apt-get update*
- utiliser un compte système : *sudo* ***commande*** **|** *sudo su*
- avant tout paramétrage, faire une **copie de sauvegarde** des fichiers de configuration qui seront modifiés

Après toute modification, vous devez **recharger le service** (*service nom_service reload*) ou le redémarrer complètement ce qui interrompt les connexions établies sur le serveur (*service nom_service restart*).

UTILISATION DU *SHELL*

Environnement *shell*
Sous Linux, l'utilisation de la ligne de commande se fait dans un environnement riche (le *shell*) qui propose des fonctions de **complétion** (il complète) grâce à la touche tabulation.

Mode administrateur
Les commandes d'administration qui doivent être exécutées en mode administrateur seront préfixées par la commande *sudo*.
On peut aussi passer complètement en mode *superutilisateur* en tapant *sudo su*. Il y a un risque car les manipulations opérées le sont sans demande de confirmation.

Lancement d'un processus autonome
Lorsqu'on lance un processus (par exemple un éditeur) depuis le *shell*, ce processus *fils* prend la main et on ne pourra utiliser le *shell* qu'après avoir fermé le processus *fils*.

On peut lancer un processus autonome déconnecté du *shell* en finissant la commande par *&* :
root@monLinux:/# gedit /etc/resolv.conf &

Aide sur une commande
On obtient de l'aide ponctuelle en utilisant l'option *--help* (ex: *mkdir --help)* ou de l'aide détaillée en consultant le manuel (*man nomCommande*).

ARBORESCENCE DES FICHIERS

Contrairement à Windows, Linux ne propose pas un ensemble de lettres de lecteur pour les différentes partitions.
À la place, on trouve une arborescence unique (/) à laquelle sont rattachés les différentes partitions, lecteurs physiques (disquette, CDROM, etc) et lecteurs USB.
Pour qu'un lecteur soit visible dans cette arborescence, il faut qu'il y ait été raccroché (***monté***) par la commande *mount.* Ce terme de *monter* un lecteur vient de l'époque où il fallait d'abord installer physiquement le disque avant de le mettre en route.
L'outil de navigation en interface graphique est nommé *nautilus*.
Pour manipuler les droits et propriétaires, on peut l'utiliser en mode *superutilisateur*.

L'arborescence

On trouvera principalement dans cette arborescence :

Lieu	Contenu	
/etc	programmes et fichiers de configuration principaux	
	/init.d	Endroit où se trouvent les principales applications (un peu équivalent au Program Files de Windows)
/dev	drivers de périphériques	
/media	Montage des répertoires pointant sur des unités de stockage et lecteurs disque amovibles Apparaît parfois sous le nom /mount ou /mnt dans d'autres distributions	
/home	Stockage des répertoires de travail des utilisateurs (équivalent de c:\Documents and settings)	
/bin	Contient les principales commandes en ligne	
/var	Fichiers et données variables (données dynamiques, fichiers temporaires, pages web, etc) pour les applications	
/usr	Fichiers de configuration propres à la session de l'utilisateur	

Navigation dans l'arborescence

Voici quelques commandes pour naviguer dans les répertoires et explorer le système de fichiers :

Action	Commande	Options
Afficher le contenu d'un répertoire	*ls*	-l : affiche l'utilisateur et le groupe propriétaires ainsi que les droits d'accès -R : affiche le contenu du répertoire et des sous-répertoires -all : comme -l, y compris pour les fichiers cachés
Créer un répertoire	*mkdir*	*nomRepertoire*
Supprimer un répertoire	*rmdir*	*nomRepertoire* Le répertoire doit être vide
Supprimer un fichier	*rm*	*nomFichier*
Se déplacer dans l'arborescence	*cd*	/ retourne à la racine .. remonte d'un niveau dans l'arborescence nomRepertoire descend à l'intérieur de l'arborescence

Fichiers, droits d'accès, propriétaire

Commandes pour définir ces autorisations et déterminer le propriétaire d'un fichier ou répertoire :

action	commande	remarques
Visualiser les droits d'un élément	*ls -l* *ls -all*	Affiche les fichiers courants Affiche les fichiers courants et système (cachés)
Modifier le propriétaire (change owner)	*chown*	Usage : chown nom_utilisateur nom_fichier
Modifier les droits	*chmod*	Usage : chmod *nvx_droits nom_fichier* les *nvx_droits* sont une valeur numérique de trois nombres représentant en binaire les droits pour l'utilisateur, son groupe et les autres. Par exemple : 7= $(111)_2$: tous les droits (RWX) 4 = $(100)_2$: droit de lecture seule (R~~WX~~)

CONFIGURATION RESEAU

1 Configuration de l'adressage statique (IP fixe)

La configuration de l'adresse IP, du masque ou l'activation / désactivation de la carte réseau se fait par la commande ifconfig
ifconfig *nom_carte [adresse_ip] [**netmask** valeur_masque] [**up**|**down**]*

2 Configuration de l'adressage dynamique (DHCP)

Pour obtenir une adresse IP d'un serveur DHCP, ou pour renouveler un bail, on utilisera la commande ***dhclient*** nom_carte

3 Configuration du routage

La configuration du routage (et donc de la passerelle par défaut pour un poste) utilise la commande *route*.

- Ajouter une ligne dans la table

route add adresse_réseau ***[netmask*** valeur_masque***] [gateway*** adresse_passerelle***] [metric*** valeur***]***

- Ajout d'une route par défaut (donc d'une passerelle)

route add default [netmask valeur_masque***] [gateway*** adresse_passerelle***] [metric*** valeur***]***

- Supprimer une ligne dans une table

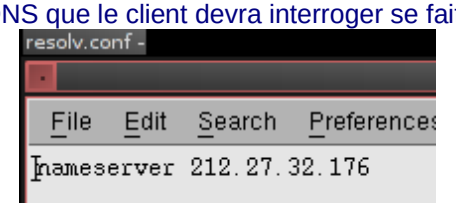
route del adresse_réseau ***[netmask*** valeur_masque***] [gateway*** adresse_passerelle***] [metric*** valeur***]***

- Afficher la table de routage

route –e

4 Configuration du client DNS

La configuration des serveurs DNS que le client devra interroger se fait dans le fichier /etc/resolv.conf



On y ajoute les adresses des serveurs sous la forme ci-contre.

PAQUETAGES ET SERVICES

Gestion de paquetages
L'installation d'applications système (DNS, DHCP, serveur Web, etc) sous Linux nécessite les droits administrateur.
La commande « ***apt-get*** » est le gestionnaire de ***paquetages*** (on pourra aussi utiliser l'équivalent ***aptitude*** ou le logiciel en interface graphique ***synaptic***).
Il permet diverses manipulations selon les options utilisées, parmi lesquelles :

- ***update*** : met à jour les dépôts de paquetages
- ***install*** *paquetage1[paquetage2 paquetage3 …]* : installer un ou plusieurs paquetages
- ***remove*** *paquetage1[paquetage2 paquetage3 …]* : supprime un ou plusieurs paquetages. Les fichiers ne sont pas supprimés, y compris les configurations de l'installation
- ***autoremove*** : supprime les paquetages installés automatiquement et qui n'ont plus d'utilité, ou les éléments supprimés par l'option *remove* et qui sont encore présents dans le système

Gestion des services
Une fois les paquetages installés, les applications systèmes (ou services) présents dans ***/etc/init.d*** peuvent être :

- démarrés : ***start***
- arrêtés : ***stop***
- re-démarrés (ils sont d'abord arrêtés, puis démarrés) : ***restart***
- leurs fichiers de configuration peuvent être rechargés (évite l'interruption du service pour les utilisateurs déjà connectés) : ***reload***

Exemple :
 # /etc/init.d/apache2 restart
 # /etc/init.d/bind9 reload

On peut aussi connaître le statut (running, failed, stopped, etc) d'un service en utilisant l'option ***status***:
 # /etc/init.d/mysql status

Beaucoup de services peuvent être gérés avec la commande ***service*** :
 # service apache2 restart
 # service bind9 reload
 # service mysql status

UTILISATION DES FICHIERS DE CONFIGURATION

Contrairement à Windows où les configurations des services se passent par interface graphique et sont stockées dans la base de registres, Linux utilise encore des fichiers texte.
Voici quelques éléments pour comprendre leur fonctionnement et utiliser au mieux une approche modulaire.

Contenu/Syntaxe
Un fichier de configuration est éditable à partir d'un utilitaire en mode texte (***vi, vim, nano, pico*** ou autre, voir annexe pour quelques commandes) ou un éditeur plus complet si on dispose d'une interface graphique (par exemple ***gedit*** sous Ubuntu).
Les éléments de syntaxe suivants sont très courant

Elément	*Usage*
#	Commentaire textuel qui donne des explications sur les options
;	Commente une ligne de paramétrage. A dé commenter si on veut activer l'option
<intitulé> …. </intitulé>	Encadre une description complexe (même principe que pour les balises HTML ou la structure d'enregistrement en algorithmique)
	On sépare différentes valeurs par des espaces

Approche modulaire
Pour éviter d'encombrer un fichier de configuration, on a recours à une externalisation (comme on le fait pour le CSS qui sort la mise en page du HTML).
On crée donc des fichiers externes qui ne prennent en compte qu'une partie de la configuration, et on y fait appel dans un fichier central grâce à la clause ***include***

Exemple pour apache2.conf
 # inclue les fichiers de configuration contenus dans le sous-dossier conf.d
include conf.d/
 # inclue les déclarations d'hôtes virtuels contenus dans le sous-dossier sites-enabled
include sites-enabled/
 # Inclue la liste des ports d'écoute contenue dans le fichier ports.conf
include ports.conf

Exemple pour DNS
Le fichier de base du service Bind est ***named.conf*** dont le contenu est :
include "/etc/bind/named.conf.options"; *#options de configuration du serveur*
include "/etc/bind/named.conf.local"; *#zones gérées par le serveur*
include "/etc/bind/named.conf.default-zones"; *#zones par défaut (localhost, résolution inverse, etc)*

Nettoyage des fichiers et Versionning
Pour rendre la lecture des fichiers de configuration aisée, on épure leur contenu en supprimant toute option non utilisée, tout commentaire inutile
De manière à pouvoir revenir à la situation initiale, on a coutume de faire une copie d'un fichier de configuration avant de le modifier.
On utilise la commande ***cp*** pour copier un fichier vers un double avec un nouveau nom.
cp /etc/apache2/apache2.conf /etc/apache2/apache2.old1

EDITEUR VI

Pour manipuler les fichiers systèmes, qui sont des fichiers texte, il est nécessaire de recourir à un éditeur de texte.
Même quand aucune interface graphique n'est disponible (et les linuxiens purs de durs diront qu'en toute circonstance...), un outil rudimentaire bien qu'offrant d'innombrables possibilités permet toute manipulation sur un fichier texte : l'éditeur *vi*.
Le mode commande (auquel on revient en appuyant Echap) propose diverses actions dont les plus utiles sont présentées ci-dessous :

action	commande	remarques
insérer du texte	i	Passe l'éditeur en mode insertion. Taper à nouveau i (ou Echap) pour repasser en mode commande **attention** : les flèches sont désactivées pour les déplacements
supprimer une ligne	dd	Ces commandes ne marchent pas si on est en mode insertion
supprimer le caractère sous le curseur	x	
supprimer le caractère à gauche du curseur	X	
enregistrer les modifications	: w	
quitter vi	:q	
enregistrer et quitter	:wq	
quitter sans enregistrer	:q!	Il n'y a pas d'invite à confirmer
Rechercher du texte vers le bas du fichier	/texte_cherché	On tapera à nouveau / pour chercher la prochaine occurrence
Rechercher du texte en arrière	?texte_cherché	On tapera à nouveau ? pour chercher l'occurrence précédente

Des versions un peu plus riches sont disponibles, qui proposent des menus accessibles à partir du clavier : vim, nano, pico.

FICHE 2 : DNS

PRESENTATION

Le service DNS est le cœur des applications internet / intranet. En fournissant la possibilité d'utiliser des noms symboliques (FQDN - Fully Qualified Domain Name) à la place d'adresses IP, il a permis le déploiement de multiples applications accessibles de manière souple pour des utilisateurs. Son système repose sur une **structure arborescente**

- partant d'une racine (« . »),
- se déclinant en **domaines de référence** rattachés à cette racine (**TLD – Top Level Domain** : .fr, .com, .net...)
- détaillée en **domaines privés** (ac-caen, google, laposte...) rattachés à ces TLD
- enregistrant des **associations** entre des noms de machines dans un domaine (**www** par exemple) et l'adresse IP correspondante

PROCEDURES

Installation du service et création de la zone sur le principal

1. Installer le paquetage ***bind9***
2. **Déclarer** la zone à gérer dans ***named.conf.local*** avec le type « master »
3. **Configurer** le fichier de la zone en y ajoutant les enregistrements nécessaires (NS, A, ...)
4. Vérifier la validité de la configuration du service : ***named-checkconf***
5. Vérifier la validité de la configuration de la zone : ***named-checkzone***
6. Recharger le fichier de configuration ou redémarrer le service
7. Paramétrer un client pour qu'il utilise ce serveur, faire un ping sur un FQDN

Liste des fichiers (dans /etc/bind)

Fichier	*Usage*
named.conf	Fichier de base, renvoie vers les fichiers de configuration
named.conf.default-zones	Zones par défaut (local host, résolution inverse, etc) : décrit les zones (nom, type, fichier de configuration, etc)
named.conf.local	Fichier dans lequel déclarer les zones gérées par le serveur
db.****	Fichiers de configuration des zones (SOA, NS, A)

Installation du secondaire

1. Installer le paquetage ***bind9***
2. **Déclarer** la zone dans ***named.conf.local*** avec le type «slave»
3. Vérifier la validité de la configuration du service : ***named-checkconf***
4. Recharger le fichier de configuration ou redémarrer le service
5. Paramétrer un client pour qu'il utilise ce serveur secondaire, faire un ping sur un FQDN

MODE OPERATOIRE

CONFIGURATION SERVEUR DNS

Un système DNS repose principalement sur trois modes de serveur :

- Le **serveur autorité** : il héberge le **fichier décrivant la zone** (exemple : rostand.fr) et est le seul habilité à le modifier. On parle de serveur principal ou serveur maître.
- Les **serveurs secondaires**, qui reçoivent les mises à jour du fichier de zone, réalisant une tolérance de panne et une répartition de charge. On parle de secondaires ou esclaves.
- Les **serveurs délégués** : ils sont les hébergeurs faisant autorité sur des sous-zones (exemple btsinfo.rostand.fr). Ils doivent être déclarés au niveau supérieur (rostand.fr).

Dans un **fichier de zone**, on va trouver les **enregistrements** correspondant :

- à chaque machine référencée (A),
- aux serveurs de noms et serveurs délégués (NS),
- au(x) serveur(s) de messagerie (MX),
- au routeur (RT),
- etc.

Sous Linux, l'outil responsable de la gestion de DNS est BIND (version 9 à ce jour), qui installe le service ***named***.

1 Serveur Autorité

On parlera d'un maître, d'un primaire ou d'un principal.
Il est définit comme **Start Of Authority (SOA)**.
Il n'y a **qu'un serveur SOA** pour un domaine DNS. Il est autorisé à apporter des modifications dans le fichier de zone.
Il comporte l'ensemble des informations nécessaires à faire fonctionner BIND. À savoir :

- un fichier de configuration du service et de déclaration des zones (/etc/bind/named.conf complété par d'autres)
- autant de fichiers de zones que nécessaires, accessibles dans /etc/bind/.

Les fichiers de configuration et la description des zones

named.conf.options
Ce fichier de référence décrit les options courantes du serveur (**attention, les valeurs sont des exemples**) :

```
options {
        directory "/var/cache/bind"; #repertoire de stockage des fichiers de zone
        dump-file "/var/data/cache_dump.db"; #repertoire de cache
        statistics-file "/var/bind/named_stats.txt"; #repertoire de statistiques
}
```

named.conf.local
On écrira dans ce fichier la **déclaration des zones** directes (sens FQDN→IP) et inverses (sens IP→FQDN). **On portera une attention particulière aux points-virgules.**

- pour chaque zone qu'il gère (exemple pour une zone nommée rostand.fr)

```
zone "rostand.fr" IN {
        type master; #serveur maître
        file "base.rostand.dns"; #nom du fichier qui décrit la zone
        allow-update { any; }; #possibilité de mise à jour en réseau vers les secondaires
};
```

- chaque zone de résolution inverse (exemple pour le réseau 192.168.0.0/24)

```
zone "0.168.192.in-addr.arpa" IN { #adresse IP du réseau à l'envers suivi de in-addr.arpa
        type master;
        file "rostand.inverse";
        allow-update { none; };
};
```

Les fichiers de zone et de zone inverse

On trouvera ensuite, conformément à ce qui a été indiqué dans ***named.conf.local***, autant de fichiers que de zones ou de zones inverse.
Un fichier de zone comporte les éléments suivants :

Enregistrement	*Rôle*
SOA	Définit les indications du Start Of Authority : nom du domaine (ou de la zone) nom de la machine qui est SOA dans ce domaine nom de l'administrateur du domaine numéro de version de fichier délais pour la synchronisation
NS	Déclare les noms des machines qui sont serveur de noms (principal ou secondaires) pour la zone Remarque : Ces noms devront en plus être associés à une adresse par un enregistrement A.
A	Déclare les associations entre FQDN et adresse IP. On parle d'un hôte **Remarques :** un nom non terminé par un point est complété par la zone décrite dans le SOA un nom terminé par un point est un FQDN
MX	Déclare le nom de la ou des machines assurant la fonction de serveur de messagerie pour le domaine. **Remarque :** Ces noms devront en plus être associés à une adresse par un enregistrement A.
RT	Déclare le nom de la ou des machines assurant le rôle de routeur dans le domaine. Utilisé pour les systèmes avec auto-configuration. **Remarque :** Ces noms devront en plus être associés à une adresse par un enregistrement A.
PTR	Enregistrement inverse qui associe le nom FQDN à une adresse IP de machine dans le réseau IP déclaré dans le SOA. Utilisé pour les systèmes de cartographie de réseau ou pour l'administration distante

Exemple

```
#Fichier rostand.dns décrivant la zone rostand.fr
#SOA : start of authority + nom de zone + nom de l'administrateur
@ IN SOA rostand.fr root.rostand.fr (
        42 ; numéro de série important pour les secondaires (actualisé à chaque modification)
        3H ; temps de rafraîchissement des secondaires (3 herues)
        15M ; temps d'attente entre deux tentatives de mise à jour pour les secondaires (15 min)
        1W ; durée de vie d'une information (1 week)
        1D ) ; temps avant la déclaration d'invalidité permanente du principal (1 day)
IN NS dns.rostand.fr. ; déclaration serveurs de noms principaux et secondaires
btsinfo NS srvinfo.rostand.fr. ;délégation d'autorité pour la sous-zone btsinfo.rostand.fr
                               ; le serveur aura pour nom srvinfo.rostand.fr
MX 10 smtp ; pointeur pour le serveur de messagerie avec numéro d'ordre
MX 20 mail ; deuxième pointeur, serveur secondaire
dns A 192.168.0.152 ; association pour le nom de machine dns.rostand.fr
www A 192.168.0.152 ; déclaration d'association pour le nom de machine www
srvinfo A 192.168.0.153 ; association pour la machine srvinfo
smtp IN A 192.168.0.253 ; association pour le nom smtp
mail IN A 192.168.0.154 ; association pour pour le nom mail
console CNAME srvinfo ;alias pour le nom de machine srvinfo
www6 IN AAAA ::1 ;association pour une adresse IPv6
```

Et pour la résolution inverse :

```
#Fichier rostand.inverse décrivant la zone 0.168.192.in-addr.arpa
#SOA : start of authority + nom de zone + nom de l'administrateur
@ IN SOA 0.168.192.in-addr.arpa root. 0.168.192.in-addr.arpa (
42 3H 15M 1W 1D )
IN NS rostand.fr ; déclaration serveurs de noms par nom DNS

152 IN PTR www ; association le numéro 253 vers le nom de machine www
153 PTR srvinfo ; association pour le numéro 153 vers la machine srvinfo
253 PTR smtp ; association pour le numéro 253 vers le nom smtp
```

2 Serveur secondaire ou esclave

Un serveur secondaire est ni plus ni moins qu'un serveur BIND qui a déclaré dans son fichier ***named.conf.local*** qu'il était *esclave* pour une zone déterminée (la terminologie Windows est *secondaire*).

```
zone "rostand.fr" IN {
        type slave; #serveur esclave
        masters {192.168.0.152 ;} #adresse des serveurs maîtres
        file "double.rostand.dns"; #nom du fichier si on veut en conserver une copie en local
        allow-update { none; }; #impossibilité de mise à jour en réseau vers d'autres secondaires
};
```

Dès lors, il recevra à intervalle régulier les mises à jour.
Il est important de changer le numéro de version à chaque modification du fichier principal de sorte que les secondaires se mettent correctement à jour.

3 Serveur de zone délégué

Un serveur de zone délégué est un serveur DNS qui a l'autorité sur un sous ensemble d'une zone principale. Par exemple, la zone .gouv.fr. a délégué la responsabilité de nombreuses sous-zones : education.gouv.fr, impots.gouv.fr, etc.
Pour chaque sous-zone, une inscription a été faite dans le fichier de zone principale, et chaque délégation renvoie vers un nouveau serveur SOA pour la sous zone.
Cet enregistrement au niveau supérieur est indispensable pour éviter tout ajout non désiré (on ne peut pas décider librement de créer une sous-zone, il faut qu'on nous ait donné une délégation).

Exemple, dans le fichier de la zone gouv.fr, on trouvera :

```
education IN NS nom_dns_du_serveur_délégué_education
impots IN NS nom_dns_du_serveur_délégué_impots
```

puis

```
nom_dns_du_serveur_délégué_education IN A adresse_ip_education
nom_dns_du_serveur_délégué_impots IN A adresse_ip_impots
```

Pour le serveur délégué, il s'agit ni plus ni moins qu'un maître pour la zone education.gouv.fr ou impots.gouv.fr.

4 Test des configurations

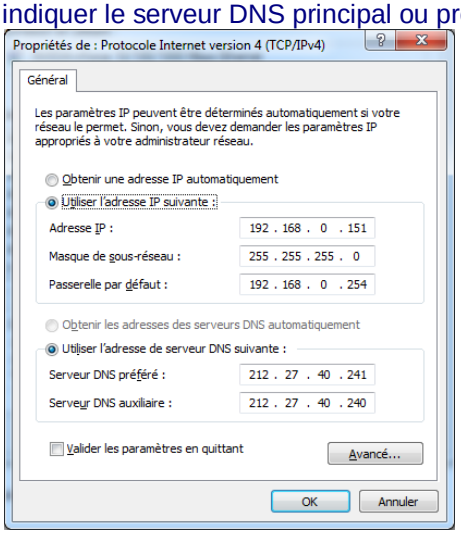
Avant de lancer un serveur suite à une modification, on peut prendre la précaution de tester les configurations des fichiers.

Commande	Rôle
named-checkconf	teste la validité des déclarations de zone (fichier named.conf et fichier de déclaration named.conf.local, named.conf.default-zones, etc).
named-checkzone	Teste la validité d'une zone à partir de son fichier de configuration named-checkzone nomZone cheminFichierZone

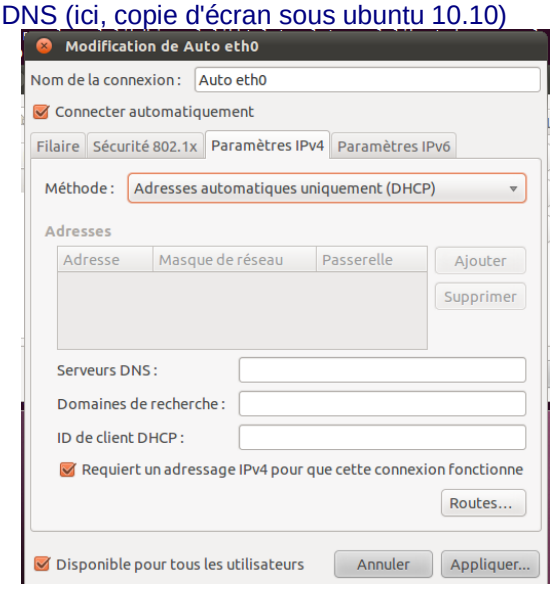
CLIENT ET TEST

On paramètrera les clients pour qu'ils aillent s'informer auprès de leur serveur DNS (qui peut être le secondaire ou le principal).

Sous Windows
Dans les propriétés de la carte, indiquer le serveur DNS principal ou préféré



Sous Linux
Dans l'environnement graphique, utiliser « Adresses Automatiques uniquement » pour donner une valeur manuelle à la partie DNS (ici, copie d'écran sous ubuntu 10.10)



Test

On teste le bon fonctionnement du serveur DNS en tapant :

- *ping nomFQDN : prouve que la résolution fonctionne*
- *ping -a adresseIP : si la résolution DNS inverse fonctionne, donne le nom FQDN de la machine possédant l'adresse interrogée*
- *nslookup nomFQDN : interroge le serveur DNS pour connaître l'adresse IP correspondant au nom FQDN*
- *nslookup adresseIP : interroge le serveur DNS pour savoir si un enrtegistrement inverse est associé à l'adresse IP et récupère alors le FQDN correspondant*
- *nslookup -type=champ nomDomaine : interroge le champs particulier (NS, MX, RT, etc) pour le domaine spécifié*
- *dig @adresseServeur nomzone champ (sous Linux) : permet d'interroger le contenu d'une zone en demandant les informations associées à un champ particulier (NS, MX, RT, etc)*

Source

- http://www.zytrax.com/books/dns : site complet en anglais sur DNS et Bind9

FICHE 3 : SERVEUR WEB APACHE

PRESENTATION

Serveur HTTP
Un serveur Web est une machine hébergeant le service réseau HTTP (HyperText Transfert Protocol).
Le protocole HTTP permet au serveur de retourner, à la demande d'un client :

- Les **entêtes** : nom du serveur, versions du protocole supportées, autres fonctions accessibles...,
- Des **pages HTML** et les éléments qui leurs sont liés (CSS, scripts côté client, images appelées par <img>, modules incorporés par <embed> comme du ***flash*** ou du ***streaming***, etc)
- Des pages HTML résultant de **l'exécution de pages de script côté serveur** (PHP, ASP, JSP,...).

Apache
Le serveur Web le plus utilisé dans le monde (source ***journal du net*** ici) est un outil libre développé dans l'environnement Linux/UNIX : le serveur Apache de l'Apache Software Foundation (www.apache.org) qui se décline en d'innombrables versions (dont Tomcat pour l'exécution de script côté serveur en Java : servlet).
Il permet la mise en place d'un serveur web (pages HTML) avec exécution de scripts PHP, JSP ou ASP.
Il peut sécuriser limiter l'accès aux pages (.htaccess), créer des hôtes virtuels (plusieurs noms de site différents sur une seule exécution du service) ou des sites indépendants (sur des ports différents).
L'installation du service standard ne permet pas la sécurisation SSL.

PROCEDURES

Installation du service

1. Installer des paquetages ***apache2*** (et, pour les versions d'Ubuntu avant 10.04, installer manuellement les paquetages ***apache2.2-common*** et ***apache2-utils***) pour la version de base
2. Pour ajouter des fonctions d'authentification, la gestion du multi-processing et la possibilité de changer le service en mode ***root*** : installer les paquetages ***apache2-mpm-prefork, libapache2-mod-chroot, libapache2-mod-auth-pam, libapache2-mod-auth-sys-group***
3. Tester depuis une machine cliente (http://adresseIPServeur)

Le répertoire par défaut des fichiers HTML et PHP est ***/var/www***.
Les principaux fichiers de configuration (dans ***/etc/apache2***)

Fichier	*Usage*
apache2.conf	Définit les principales caractéristiques techniques du service HTTP : mode de transfert de l'information, durée avant la fermeture d'une session, gestion de la sécurité, etc. Fait appel aux fichiers externes pour le reste de la configuration
ports.conf	Indique les ports sur lesquels le serveur écoute (défaut HTTP:80, HTTPS:443). Permet la création d'hôtes virtuels (un seul serveur apache, plusieurs sites sous des noms différents)
httpd.conf	C'est l'ancien fichier de configuration. Il peut être intéressant d'y insérer les paramétrages spécifiques pour améliorer la lisibilité (vérifier qu'il est appelé par « include » dans ***apache2.conf***)

Installation du PHP
Pour prendre en charge PHP, Apache doit être complété par un interpréteur.

1. Installer le module ***php5***, qui installera les dépendances ***php5-common, php5-gd, php5-cli*** et ***libapache2-mod-php5***
2. Redémarrer le serveur apache
3. Tester en déposant ou créant un fichier avec du code PHP dans le répertoire ***/var/www***

Prise en charge MySQL par Apache/PHP
En complément du serveur Web, on installe la capacité pour Apache de gérer les échanges avec MySQL.
Cela ne nécessite pas nécessairement que MySQL soit présent sur la machine hébergeant Apache (MySQL peut être sur un serveur Windows sous la forme d'une installation EasyPHP par exemple)

1. Installer le paquetage ***libapache2-mod-auth-mysql*** pour la gestion de l'authentification MySQL
2. Installer le paquetage ***php5-mysql*** pour permettre les échanges entre PHP et la base MySQL
3. Redémarrer apache
4. Tester en insérant une connexion à une base MySQL dans un fichier PHP sur ***/var/www***

Remarque : pour des raisons évidentes de sécurité, le compte « root » n'est pas autorisé à établir des connexions à la base depuis d'autres machines que ***localhost***.
On devra donc procéder comme suit sur le SGBD

1. Créer un utilisateur MySQL avec mot de passe (***create user nomutil identified by 'motPasse' ;***)
2. Lui donner le droit de se connecter depuis toute machine (***grant usage on *.* to 'nomutil'@'%'***)
3. Lui donner éventuellement les droits nécessaires sur la base de données particulière

MODE OPERATOIRE

SERVEUR WEB, REPERTOIRES VIRTUELS, SITES MULTIPLES

1 - Installation et répertoires

Un serveur web correspond à un service en écoute sur le port 80 (port standard pour le protocole HTTP).
L'installation d'Apache (***apt-get install apache2***) entraîne l'installation du service ***/etc/init.d/apache2***.
Le fichier principal de la configuration est ***/etc/apache2/apache2.conf*** qui fait lui-même appel à d'autres fichiers externalisés.
Les pages Web sont stockées par défaut dans ***/var/www***.

2 - Fichier apache2.conf

Généralités
Ce fichier comporte dans sa version standard plus de 200 lignes (dont beaucoup de commentaires).
Il est conseillé de travailler sur une copie du fichier d'installation dans lequel on supprimera tous les éléments inutilisés.
Bien entendu, il est prudent de savoir ce que contiennent les directives avant de les activer, supprimer ou modifier, comme cela est rappelé dans les premiers commentaires :

```
# Do NOT simply read the instructions in here without understanding
# what they do. They're here only as hints or reminders. If you are unsure
# consult the online docs. You have been warned.
```

Remarque : Les commentaires du fichier source présenté plus loin ont été supprimés, seules les directives principales et basiques sur lesquelles on intervient en général sont présentées.

Structure du fichier
Il y a trois grandes sections dans ce fichier :

1. **globale** : définit les paramètres du service (durée de session, connexions simultanées, etc)
2. **site standard** (fichiers dans ***httpd.conf***) : paramétrage manuel du service standard (répertoires, droits d'accès, etc)
3. **hôtes virtuels** (fichiers dans ***sites-enabled/***) : paramétrage de sites accessibles sous des noms ou adresse IP virtuelles

Section 1 : Paramétrage du service

Dans cette première section, on paramètre la façon dont le service HTTP prendra en charge les demandes de connexion.
On y précise aussi les inclusions pour ce qui est du reste de la configuration (cela évite d'avoir un fichier unique de 900 lignes comme le ***httpd.conf*** de la version 1 de Apache).

Contenu du fichier (Extraits)

Timeout 300 *; durée de vie d'une connexion en secondes (temps avant la déconnexion automatique)*
KeepAlive On *; On ou Off : permet pour une même connexion de faire plusieurs demandes.*
; améliore les temps de réponse, oblige le serveur à garder des sessions en mémoire
KeepAliveTimeout 15 *; temps en seconde entre deux actions avant que la session soit déconnectée*
AccessFileName .htaccess *; indique la façon dont on restreint l'accès aux dossiers*
; (ici, fichiers .htaccess dans le dossier à sécuriser
; à commenter si on ne veut pas appliquer de restrictions
LogLevel warn *; indique le type d'événement enregistré dans les journaux*
; valeurs : debug, info, notice, warn, error, crit, alert, emerg.
Include mods-enabled/*.load *;modules appelés au chargement du service (authentification, PHP,*
; transferts de fichiers riches type PDF, MP3 et autres...)
Include mods-enabled/*.conf *;configurations spécifiques pour les modules*
 # Include all the user configurations:
Include httpd.conf ;*appel aux configurations spécifiques (section 2 : « service standard »)*
 # Include ports listing
Include ports.conf *;appel aux paramétrages sur les ports d'écoute*
 # Include generic snippets of statements
Include conf.d/ ; *appel au dossier contenant les paramétrages spécifiques des sites*
; par exemple, on y trouve phpmyadmin.conf
 # Include the virtual host configurations:
Include sites-enabled/ *; sites gérés, construit lors du lancement du service à partir des fichiers*
; de configuration

Fichier *ports.conf*

NameVirtualHost *:80 *;Gestion du serveur avec prise en charge d'hôtes virtuels sur le port 80*
Listen 80 *; port d'écoute (80 est la valeur par défaut).*
<IfModule mod_ssl.c> *; paramétrage si prise en charge de SSL (remplacé par TLS)*
Listen 443
</IfModule>
<IfModule mod_gnutls.c> *; paramétrage si prise en charge de TLS*
Listen 443
</IfModule>

Section 2 : Configuration du site Web

On trouvera dans cette section les directives pour le site principal (c'est à dire pour une installation sans hôte virtuel).
À cet endroit, on définit les caractéristiques du site, les sous répertoires accessibles, et les options de sécurité qui s'y appliquent.
On peut mettre ces directives dans le fichier ***httpd.conf***, ou dans le fichier ***apache2.conf*** (non recommandé).
En cas d'utilisation d'hôtes virtuels, il est préférable de renseigner ces informations dans le fichier ***default.conf*** du sous-dossier ***sites-enabled***.

Principales directives

Directive	Usage
<Directory...> </Directory>	Définitions de conditions et comportements particuliers pour un répertoire (chemin d'accès, restrictions d'accès, etc)
Alias	Définit un raccourci pour un chemin d'accès complexe à un répertoire
Allow	Autorise l'accès à machines ou des adresses de réseau
Deny	Interdit l'accès à des machines ou des adresses de réseau
DocumentRoot	Chemin d'accès à un dossier contenant des pages Web
LogLevel	Niveau d'enregistrement des problèmes dans le journal d'événement
Order	Ordre dans lequel sont appliquées les règles de sécurité (Deny et Allow)
ServerAdmin	Nom ou adresse de l'administrateur du serveur
ServerName	Nom du serveur tel qu'il devra être tapé dans la barre d'adresse du navigateur N'a d'intérêt au niveau général que si on gère des hôtes virtuels sur le serveur

Exemple de fichier (Extraits)

```
ServerAdmin webmaster@gsb.net ; pour joindre l'administrateur
ServerName public.gsb.net ; nom FQDN auquel répond le serveur
DocumentRoot /var/www ; localisation des pages
<Directory /> ; droits s'appliquant à la racine (définie dans DocumentRoot) du site
Options FollowSymLinks ; autorise l'usage de liens symboliques (non maîtrisé à ce jour)
AllowOverride None ; les droits ne peuvent être redéfinis ailleurs (dans fichiers .htaccess)
Deny from All ; interdit le parcours du répertoire
</Directory>
<Directory /var/www/> ; paramétrage pour le répertoire /var/www
Options Indexes FollowSymLinks MultiViews
AllowOverride None
Order allow,deny ; ordre de consultation des droits
allow from all ; autorise l'accès à toute machine
</Directory>
Alias /doc/ "/usr/share/doc/" ; raccourci pour le répertoire
<Directory "/usr/share/doc/"> ; paramètres spécifiques
Order deny,allow
Deny from all
Allow from 127.0.0.0/255.0.0.0 ::1/128
</Directory>
```

L'accès n'est autorisé que depuis la machine locale (en IPv4 et IPv6)

Section 3 : Hôtes virtuels

Principes

La configuration standard (dans ***httpd.conf***) ne permet d'avoir, pour un serveur apache, qu'un seul site web.
On peut choisir d'utiliser un serveur physique différent (ou autant de machines virtuelles) pour chaque site hébergé.
On peut aussi décider de gérer des sites accessibles par des noms différents (noms FQDN ou adresse IP) depuis un même serveur Apache.
On aura alors recours à la définition d'hôtes virtuels.

Remarque : il ne faut pas confondre un ***hôte virtuel*** et un ***sous-répertoire*** (ou ***répertoire virtuel*** dans IIS) : ce dernier est accessible par http://nom_serveur/nom_Repertoire_virtuel, comme cela est utilisé pour les "pages perso" des hébergeurs gratuits par exemple, ou pour l'accès à www.etab.ac-caen.fr/rostand. Le répertoire virtuel est géré par la directive ***Alias*** dans la configuration Apache.

Fichier *default* dans le dossier *sites-available*

Le fichier permet de créer des *hôtes virtuels* qui seront accessibles par http://nom_hote_Virtuel.
Un hôte virtuel se comporte comme un nouveau serveur web et peut donc reprendre les directives indiquées plus haut (DocumentRoot, Listen, Directory, etc.).

Exemple de déclaration d'hôtes

```
#définition d'un hôte virtuel depuis n'importe quelle adresse IP du serveur, sur le port 80
<VirtualHost *:80>
NameVirtualHost 192.168.0.152 ;Déclaration d'un hôte virtuel sur une adresse IP déterminée
DocumentRoot /var/www
</VitualHost>
#Définition d'un hôte virtuel sur une adresse précise du serveur, avec pour nom virtuel « www.SiteVitrine.fr »
<VirtualHost 192.168.0.152:80>
        DocumentRoot /var/www/SiteVitrine ; chemin où sont stockées les pages du site
        ServerName www.SiteVitrine.fr ;nom du site virtuel
#Déclaration des options de sécurité du répertoire virtuel
#Choisir la gestion par .htaccess, limiter les droits de lecture/modification, etc
</VirtualHost>

#Définition d'un hôte virtuel sur la même adresse pour un second site de nom «  commerce.online.fr »
<VirtualHost 192.168.0.152:80>
        DocumentRoot /var/www/SiteCommercial ; chemin où sont stockées les pages du site
        ServerName commerce.online.fr ;nom du site virtuel
#Déclaration des options de sécurité du répertoire virtuel
</VirtualHost>
```

L'accès à l'hôte virtuel nécessite bien entendu de pouvoir l'atteindre grâce à un FQDN, donc grâce à un enregistrement DNS ou un renseignement dans le fichier hosts ou lmhosts (Voir fiche DNS).

Écoute sur plusieurs ports

Pour permettre la mise en place de sites spécifiques ou sécurisés (parce qu'ils n'écoutent pas sur le port standard), on pourra aussi créer des hôtes virtuels en écoute sur un port spécifique.

```
#Définition d'un hôte virtuel sur une adresse précise du serveur, sur un autre port (exemple ici : port 8800)
<VirtualHost 192.168.0.152:8800>
        DocumentRoot /var/www/intranet ; chemin où sont stockées les pages du site
        ServerName intra.entreprise.fr ;nom du site virtuel
</VirtualHost>
```

SECURISATION DES ACCES (.htaccess)

Par défaut, un serveur Web ne permet pas la sécurisation des informations autrement que par les droits d'accès accordés à l'utilisateur qui exécute le processus. Il n'est donc pas possible de réaliser de sécurité en fonction des utilisateurs.
Avec Apache (mais aussi avec IIS), une connexion authentifiée va autoriser d'affiner les accès en fonction des utilisateurs grâce à un fichier nommé *.htaccess* situé dans chaque répertoire pour lequel on voudra limiter l'accès spécifiquement.
Ces fichiers définissent les utilisateurs autorisés pour le répertoire courant et tous les répertoires, jusqu'à rencontrer un nouveau fichier *.htaccess* plus bas dans l'arborescence.
Pour cela, chaque répertoire (éventuellement dans les hôtes virtuels) sera paramétré pour prendre en compte ces fichiers grâce à la directive

Exemple de fichier de recours aux fichiers .htaccess

```
<Directory "MonRepertoire">
#Déclaration des options de sécurité du répertoire
        AllowOverride AuthConfig ; Les fichiers d'authentification .htaccess s'ils existent
                                   ; remplacent les droits du dossier
</Directory>
<VirtualHost 192.168.0.152:80>
        DocumentRoot /var/www/SiteCommercial
        ServerName commerce.online.fr ; nom du site virtuel
<Directory /var/www/SiteCommercial> ; Gestion de la sécurité spécifique à l'hôte virtuel
        AllowOverride AuthConfig
</Directory>
</VirtualHost>
```

Chaque fichier .htaccess est constitué comme suit:

```
AuthType Basic ; type d'authentification standard (mots de passe en clair)
;
AuthUserFile /etc/apache2//httpUtilisateurs ; chemin et nom du fichier des utilisateurs (fichier à
créer)
AuthName "Accès réservé" ;commentaire pour les utilisateurs refusés
require user nom_utilisateur1 ;utilisateurs autorisés
require user nom_utilisateur2
```

La création des utilisateurs se fera par l'outil ***htpasswd*** comme suit :

```
#La première création de compte nécessite aussi la création du fichier
#htpasswd -c /chemin/nom_fichier_utilisateurs nom_utilisateur
htpasswd -c /etc./httpd/conf/httpUtilisateurs pierre
#Le système demandera alors les mots de passe
#Pour les suivants
htpasswd paul
htpasswd jacques
```

Sources internet

- http://www.loligrub.be/contrib/tlepoint/BASE/index.html Site (en français) très détaillé sur beaucoup de configurations dans le monde de l'administration système et réseau (ici pour Apache, avec explication assez claire des différentes options et directives)

- **http://doc.ubuntu-fr.org/lamp** : en français, installer pas à pas un LAMP sous Ubuntu

FICHE 4 : SERVEUR DE BASE DE DONNEES MYSQL

PRESENTATION

Le serveur de base de données est un outil indépendant d'un serveur Web.
Son rôle est de permettre le stockage, l'exploitation et la sécurisation de données structurées en bases de données (un nom pour un domaine de gestion) au sein desquelles seront visibles les tables et occurrences.

PROCEDURES

Installation du service

1. Installer le paquetage ***mysql-server***
2. Au cours de l'installation, penser à donner un mot de passe à l'utilisateur ***root*** (et penser à s'en souvenir)
3. Vérifier la connexion : *mysql -u nomUtilisateur [-p] [nomBaseDonnees]*
   (-p permet de saisir un mot de passe, on peut se connecter directement à la base de données voulue)

Administration par PhpMyAdmin

PHPMyAdmin est une surcouche graphique permettant d'administrer la base de données MySQL par des pages Web (en PHP)
On procèdera d'abord à l'installation d'Apache et PHP conformément à la procédure précédente.

1. Installer le paquetage ***phpmyadmin***
2. Au cours de l'installation :
   a. Choisir la configuration automatique du serveur Apache
   b. Implanter la base de données nécessaire à la gestion de PhpMyAdmin (en indiquant le mot de passe de MySQL)
   c. Configurer un mot de passe spécifique pour PhpMyAdmin
3. Vérifier le fonctionnement par le navigateur : http://localhost/phpmyadmin.
4. Se connecter en utilisant le compte de MySQL

MODE OPERATOIRE

Il ne s'agit pas ici de faire un cours sur le langage SQL mais de présenter les principales commandes d'utilisation en dehors de l'interface graphique ***phpmyadmin***.

MYSQL SANS INTERFACE GRAPHIQUE

Après l'installation du service, on intervient dans MySql par la console en ligne de commande.

Se connecter à MySQL

La commande ***mysql*** tente une connexion avec le compte *root* sans mot de passe.
Elle devrait normalement déboucher sur un échec.

Syntaxe

```
mysql -u nomUtil [-p] [nom_base_de_donnees]
```

Permet la connexion sous un nom d'utilisateur, avec demande du mot de passe.
On peut aussi directement utiliser une base de données.

Exemple

```
mysql -u uGSB -p bazGSB
```

Pour utiliser une base particulière, on a recours à la commande ***use***

```
use nom_base_de_donnees
```

Objets
Pour connaître/créer les objets d'un serveur MySQL, on a recours aux commandes suivantes :

Commande	*Explication*
SHOW DATABASES	Affiche la liste des bases de données
CREATE DATABASE *nomBase*	Crée un conteneur de base de données (il n'y a pas de tables)
SHOW TABLES	Liste les tables contenues dans une base de données
DESC *nom_table*	Présente la structure de la table *nom_table*
CREATE TABLE *nom_table*	Crée une table (il faudra aussi décrire ses champs et leurs propriétés, ainsi que les contraintes)
DROP TABLE *nom_table*	Détruit la table (structure et contenu)
ALTER TABLE *nom_table*	Modifie la structure de la table (type d'un champ, ajout de colonne, ajout de contrainte d'intégrité, etc).

Gestion des droits
Les clauses SQL de base pour la création des comptes utilisateur et la gestion des droits sont les suivantes :

Clause	*Explication*	*Exemple*
CREATE USER	Crée un compte utilisateur	CREATE USER compta IDENTIFIED BY 'mpcompta'
GRANT	Attribue des privilèges à un compte	GRANT USAGE ON *.* TO compta@'nom_machine' *//donne des droits d'utilisation sur toute table au compte* **compta** *depuis la machine* **nom_machine**
REVOKE	Retire des privilèges à un compte	REVOKE SELECT ON bdGest.employes FROM compta *//retire le droit de lire (***select***) la table* **employes** *de la base* **bdGest** *au compte* **compta**

MIGRATION DE BASE DE DONNEES

Principes
La migration d'une base de données consiste à prendre un contenu (structure et données) et à le transférer dans un environnement différent : par exemple une base existant sous Acces que l'on souhaite transférer vers un environnement Serveur améliorant les performances et la

On peut le faire de plusieurs façons. En voici quelques unes.
Remarque : Le code SQL proposé est à adapter aux possibilités de syntaxe de l'environnement.

Deux bases en parallèle, recopie à l'identique : les deux bases doivent disposer d'un environnement graphique sou Windows permettant l'utilisation d'une source de données ODBC

1. Créer une base N dans le nouvel environnement,
2. Connecter l'ancienne et la nouvelle par un lien ODBC et des tables liées
3. Procéder par SQL : ***Create table*** *maNvlleBase.maTable* ***as select * from*** *monAncienneBase.maTable*
   → les tables source et destination seront exactement identiques (Structures / Données)
4. Créer les requêtes permettant de regénérer les contraintes d'intégrité :***Alter table*** *nomTable* ***add constraint*** *typeContrainte DetailContrainte*

Deux bases en parallèles, structures différentes, insertion de données. L'environnement de l'ancienne base doit disposer d'un environnement permettant la connexion ODBC, il doit exister un pilote ODBC pour accéder à l'ancien SGBD

1. Faire du ***reverse engeneering*** sur l'ancienne base (par exemple avec AMC Designor)
2. Apporter les modifications à la structure de la base
3. Générer le script de création dans le nouvel environnement
4. Implanter la base (structure) dans le nouvel environnement grâce au script
5. Les contraintes d'intégrité seront existantes dans la nouvelle base
6. Connecter l'ancienne et la nouvelle
7. Procéder par SQL, en pensant à l'ordre de réalisation des insertions du fait des contraintes d'intégrité : ***Insert into*** *maNvlleBase.maTable* ***(****liste_des_champs****) as select l****iste des champs* ***from*** *monAncienneBase.maTable*

Script d'insertion

1. Faire du ***reverse engeneering*** sur l'ancienne base, adapter éventuellement, générer le script, pour le nouvel environnement, l'implanter
2. Sur l'ancienne base, créer un *fichier* qui contiendra les données à insérer (ordres SQL, données au format XML, fichier CSV, ...). On vérifiera dans le script que l'ordre de réalisation des insertions respecte les contraintes d'intégrité (***insert into*** *(liste champs)* ***values*** *(liste valeurs)*).
   -> le script peut être généré par des fonctions d'export du SGBD, ou par une programmation dans un langage (VBA, PHP, autre)
   -> Access ou MySQL ont des fonctions permettant de générer automatiquement ce script (en SQL, en XML ou d'autres formats)
3. Exécuter le script ou importer le fichier de données dans la nouvelle base .

FICHE 5 : SERVEUR FTP (ProFTP)

PRESENTATION

Partage de fichiers
L'informatique en réseau a permis en particulier le partage de données communes (par le réseau, par messagerie, en interaction, etc).
Lorsque ces données sont stockées sur un site central sous forme de fichiers constitués (PDF, Doc, Exe, etc), il est nécessaire de créer un ***service de partage*** qui permettra à des utilisateurs d'y ***accéder par le réseau***, de manière anonyme ou authentifiée.
Cette fonction vient en ***surcouche du SGF*** (Système de Gestion de Fichiers : NTFS, FAT, Extended, etc) du disque de la machine qui héberge les fichiers.

FTP et autres techniques
De nombreuses techniques de partage existent (NFS sous Linux, DFS entre serveurs, etc.).
En environnement Windows, la fonction de partage de fichiers se nomme ***SMB*** (Server Message Bloc). Elle a été transposée en environnement Linux sous le terme de ***Samba***, qui remplit les mêmes services qu'une machine sur OS Microsoft.
Dans l'univers TCP/IP, le protocole qui prend en charge le partage se nomme ***FTP*** (File Transfer Protocol).

Fonction de partage
La fonction de partage de fichiers stockés sur un disque dur réalise tout ou partie des éléments suivants :

- rendre l'objet accessible sur le réseau (partager)
- partager de façon masquée : il faut connaître l'existence du partage pour y accéder
- authentifier les utilisateurs
- définir de droits d'accès sur les fichiers ou dossiers, à des utilisateurs ou des groupes

Le serveur peut-être installé en tant qu'outil autonome, pour le simple partage de fichiers.
Il peut aussi être complété par une fonction d'authentification avec une base de données.

PROCEDURES

Installation

4. Installer le paquetage ***proftpd***
5. Choisir une installation « ***Indépendamment*** » (le service sera lancé seul, sans passer par le gestionnaire de service ***inet*** qui lui alloue moins de ressources)
6. Vérifier le fonctionnement à partir d'un navigateur (ou d'un client FTP, port 21) : **ftp://adresseServeur** (s'authentifier avec le compte administrateur). Vous devez voir le dossier ***/home*** de l'utilisateur.

Les principaux fichiers sont (dans ***/etc/proftpd***):

Fichier	*Usage*
proftpd.conf	Définit les principales caractéristiques techniques du service FTP : nom de la machine vu sur le réseau, durée de session, nombre de connexions simultanées, modules appelés (SGBD, sécurité, etc), répertoires partagés (pour compte anonyme ou compte authentifiés), éléments de sécurité, ports d'écoute, etc
sql.conf	Définit les options pour la mise en relation avec une base de données (MySQL ou PostGreSQL) : SGBD, mode d'authentification, compte de connexion à la base, tables utilisées
ldap.conf	Définit les options de connexion à un annuaire par le protocole LDAP : nom de connexion, domaine, localisation du serveur, etc
tls.conf	Définit les options de configuration pour l'usage d'une connexion sécurisée : certificat, forcer l'authentification, etc
modules.conf	Précise les modules à charger lors du démarrage du service : base de données, sécurité, quotas, etc
virtuals.conf	Gestion d'hôtes virtuels (un serveur, plusieurs domaines)

Configuration avec MySQL pour l'authentification

1. Si besoin, procéder à l'installation de MySQL (on peut aussi utiliser un service MySQL distant)
2. Installer le paquetage de ***proftpd-mod-mysql***
3. Créer la base pour ***proftpd*** sous MySQL, les tables ***ftpgroup*** et ***ftpuser***, le compte MySQL de connexion (voir script en Annexe de cette fiche).
4. Insérer dans la table ***ftpuser*** le compte de connexion de l'utilisateur Linux
5. Configurer le fichier ***sql.conf***
6. Appeler ***sql.conf*** dans ***proftpd.conf*** (***Include***)
7. Demander à charger les modules *sql* et *mysql* dans le fichier ***modules.conf***
8. Redémarrer le service ***proftpd***

MODE OPERATOIRE

Le monde libre propose de multiples déclinaisons du protocole FTP sous forme de services : wuFTP, ProFTP, …
Nous étudierons ici les grandes lignes de ProFTP qui a la réputation d'être un peu complexe d'administration (mais proche d'Apache), « couplable » avec de l'authentification (MySQL ou LDAP), gérant des droits de partage par utilisateur ou groupe, mieux sécurisé vis a vis des failles que ***wuFTP*** par exemple.

CONFIGURATION DE BASE

Fichier proftpd.conf

Ce fichier définit le comportement du service (modules appelés, type d'adressage IP, nom du serveur, etc), mais aussi les listes de partage que l'on met en place et les droits d'accès associés.
Il fait appel à d'autres fichiers externes pour préciser certaines valeurs.

Configuration du service

Paramétrage du service (Extraits)

```
Include /etc/proftpd/modules.conf ;appelle le fichier de modules complémentaires
                                  ; (par exemple module pour MySQL ou LDAP)

ServerName "Serveur Partage" ; nom du serveur tel qu'il est vu dans l'explorateur client
ServerType standalone ; serveur qui est lancé sans l'intermédiaire
                      ;du gestionnaire de service inet (meilleure performance)
DeferWelcome on       ;n'affiche le message d'accueil qu'après authentification
DefaultServer on      ; le serveur prendra en charge les demandes de
                      ; machines non référencées dans les configurations
TimeoutNoTransfer 600 ;temps sans transfert avant fermeture de connexion authentifiée
TimeoutStalled 600    ; temps sans transfert avant fermeture connexion
TimeoutIdle 1200      ;temps sans échange (transfert, contrôle) avant fermeture

DisplayLogin welcome.msg ; fichier qui contient le message d'accueil

DenyFilter \*.*/      ; expressions régulières refusées
                      ; ici, interdit de lister tout le contenu d'un dossier

DefaultRoot ~         ;bloque les utilisateurs dans leur dossier /home/nomUtil
                      ; on peut aussi préciser un autre répertoire pour tous

# RequireValidShell off ; autorise/empêche la connexion d'utilisateur sans shell
                      ; permet de restreindre l'accès à des utilisateurs Linux

Port 21               ; port d'écoute pour les connexions (21 par défaut)
                      ; Remarque : les transferts se font sur le port 20

MaxInstances 30       ; nombre de connexions simultanées autorisées

User proftpd          ; compte utilisateur qui lance le service (sécurité)
Group nogroup         ; groupe de l'utilisateur qui lance le service (sécurité)
AllowOverwrite on     ; permet de remplacer les fichiers partagés

TransferLog /var/log/proftpd/xferlog      ; répertoire des journaux liés aux pb de transfert
SystemLog /var/log/proftpd/proftpd.log    ; journaux des problèmes système (connexion, auth...)

#Include /etc/proftpd/virtuals.conf       ; pour gérer les hôtes virtuels dans un fichier externe
```

Configuration des partages

Répertoire de connexion

Par défaut sur ProFTPd, les utilisateurs se retrouvent à la connexion dans le dossier ***/home/nomUtilisateur***.
On peut le redéfinir par la directive ***DefaultRoot*** qui bloque en outre l'utilisateur et l'empêche de remonter l'arborescence (on dit que l'utilisateur est ***chrooté***). On utilisera la valeur « ~ » qui correspond à ***/home/nomUtilisateur***, ou un dossier que l'on a choisi (par ex. /var/www pour les pages Web d'Apache).

Le dossier de connexion est défini par DefaultRoot :

- globalement pour le serveur (soit directement, soit dans une section <Global>)
- pour une connexion anonyme (<Anonymous ...>)
- dans un hôte virtuel (<Virtual Host>, voir fiche Apache).

Attribution de droits

La limitation des droits d'utilisation des dossiers s'effectue par la directive Limit :

```
<Limit liste_actions>
        Order Allow Deny            ; ordre de traitement des droits
        AllowUser monCompte         ; autorise l'accès pour l'utilisateur monCompte
        DenyAll                     ; refuse l'accès à tous les autres
</Limit>
```

Les principales actions que l'on peut restreindre par LIMIT sont décrites dans le tableau ci-dessous :

Options	Explication
ALL	Tous les droits
APPE	Ajouter du contenu à un fichier (Append) → Modifier
CWD	Se déplacer dans l'arborescence des dossiers (Change Working Directory)
DELE	Suppression de fichiers
MKD/RMD	Créer/ Supprimer des dossiers
RETR	Récupération de fichiers du serveur par le client (Download)
RNFR/RNTO	Renommer des fichiers existant (Rename From, Rename To)
STOR	Envoie de fichiers du client vers le serveur (Upload)
WRITE/READ	Écrire (créer) ou Lire des fichiers

On peut redéfinir des droits à un niveau inférieur du dossier de connexion grâce à la directive Directory :

```
<Directory /var/RessourcesBTS>      ; spécifications pour le dossier /var/ressourcesBTS
        <Limit WRITE>               ; droit d'écriture
                Order Allow Deny    ; ordre de traitement des droits
                AllowGroup gpProfs  ; réservé aux enseignants
                DenyAll             ; refuse l'accès en écriture à tous les autres
        </Limit>
        <Limit READ> ; droit de lecture
                Order Allow Deny    ; ordre de traitement des droits
                AllowAll            ; autorisé à tout le monde
        </Limit>
</Directory>
```

Exemple du fichier de base pour un partage anonyme

```
<Anonymous /home/ftp>               ; partage anonyme sur le dossier /home/ftp
        User ftp                    ; compte qui gère ce partage (compte anonyme)
        Group ftp                   ; groupe du compte

        UserAlias anonymous ftp     ; noms équivalents pour se connecter en anonyme

        RequireValidShell off       ; le compte anonyme n'existe pas sous Linux

        MaxClients 10               ; limite à 10 connexions anonymes simultanées

<Directory *>                       ; tout dossier contenu sous /home/ftp
        <Limit WRITE>               ; interdit en écriture
                DenyAll
        </Limit>
</Directory>
</Anonymous>
```

PROFTP ET AUTHENTIFICATION DEPORTEE

Par défaut, l'authentification des utilisateurs par ProFTP se fait par les comptes existant sous Linux.
Il est possible de reporter sur un système plus souple cette authentification : un fichier de comptes et groupes (non traité), un annuaire (non traité), ou une base de données.
Pour ce faire, on devra activer certains modules et renseigner certains paramètres dans les fichiers de configuration, après avoir installé les modules correspondant.

Prise en charge de l'authentification par MySQL

ProFTP doit être complété par un module de gestion de la prise en charge de l'authentification par une base de données. Pour MySQL, on installera le paquetage ***proftpd-mysql***.
On procèdera aux paramétrages ci-dessous dans les fichiers correspondant.

Fichier proftpd.conf

Dans ce fichier, on se contente d'inclure le paramétrage relatif à SQL présent dans un autre fichier.

```
#Include /etc/proftpd/ldap.conf   ; si on utilise un serveur LDAP
Include /etc/proftpd/sql.conf   ; à dé-commenter pour l'utilisation d'une base de données
```

Fichier modules.conf

Le fichier indique les modules à charger parmi ceux présents dans le dossier ***/usr/lib/proftpd***.
Dans le cas de l'utilisation d'une base de données MySQL, on dé-commentera les lignes suivantes :

```
LoadModule mod_sql.c            ; prise en charge de l'authentification par un SGBD
LoadModule mod_sql_mysql.c      ; nom du SGBD avec lequel on est en contact
```

Fichier sql.conf

C'est ici qu'on décrira le type d'authentification, le nom des tables et les champs dans lesquels on ira vérifier les informations d'authentification transmises

```
<IfModule mod_sql.c>            ;paramétrages à prendre en compte si le module SQL est activé
SQLBackend mysql                ; nom du SGBD qui contient les tables des comptes
#
#SQLEngine on
SQLAuthenticate users* groups* ; indique le niveau de droits (utilisateurs, groupes, les deux)
#
SQLAuthTypes Crypt Plaintext  ; mode de gestion des mots de passe crypté et/ou en clair

#SQLAuthTypes Backend Crypt ; si on veut que ce soit MySQL qui fasse l'authentification
#
SQLConnectInfo proftpBDD@localhost root mproot          ; coordonnées d'un compte MySQL
                                                        ;(à adapter)
#
SQLUserInfo ftpuser userid passwd uid gid homedir shell ;tables pour les comptes d'utilisateur
                                                        ; et champs étudiés
SQLUserWhereClause "LoginAllowed = 'true'"              ; n'accepte que les comptes actifs
SQLGroupInfo ftpgroup groupname gid members             ;table/champs étudiés pour groupes
#
SQLLogFile /var/log/proftpd/mysql.log ; lieu de stockage des journaux
</IfModule>
```

CLIENT FTP

L'accès à un serveur FTP doit se faire par un client :

- le ***navigateur internet***, qui est généralement limité à la récupération de contenu (mais IE permet l'envoi)
- un ***client avec interface graphique*** : type Filezilla
- la ***ligne de commande*** (ou du script)
- des ***fonctions de langage*** de programmation : voir par exemple ici pour PHP.

Connexion avec un serveur
La connexion à un serveur utilise les éléments suivants *: [utilisateur [:motpasse] @] adresse [:port]*.

Exemples :
ftp://monCompte@192.168.0.152
ftp://192.168.0.154:2121
ftp://monCompte:monPWD@192.168.0.189:4400

Eléments de la ligne de commande FTP (voir plus de possibilités sur comment ça marche)

Commande	*Utilisation*	*Exemples*
ftp	Passe en mode ligne de commande et réalise éventuellement la connexion	ftp ftp 192.168.0.156 ftp monCompte@192.168.0.152
open	Ouvre une connexion avec un serveur	open 192.168.0.156
ls	Liste le contenu d'un dossier	ls ls sousDossier ls /dossier/SousDossier
pwd	Affiche le dossier courant (Print Working Directory)	
mkd / rmd	Crée / Supprime un dossier	
put / get	Envoie / Récupère un document	put fichierChezMoi *//envoie le fichier vers le serveur* put fich1 fich2 *//envoie le fichier et le renomme* get fichierDistant *// récupère le fichier du serveur* get ficDist ficLoc *//récupère le fichier et le renomme*
!	Exécute une commande sur la machine locale	!cd .. *//se déplace dans l'arborescence locale* ! md nomDossier *///crée un dossier en local*

ANNEXE : Script pour la création de la base

Création de la base et d'un compte de connexion

```
-- création de la base de données
CREATE DATABASE proftpBDD;
-- connexion à la base
USE proftpBDD ;
-- création d'un compte MySQL pour gérer la base. Evite d'utiliser le compte root
GRANT USAGE
ON proftpBDD.*
TO proftpRoot@'localhost'
IDENTIFIED BY 'mpProftpRoot'
WITH GRANT OPTION;
```

Création des tables

```
-- Table pour les groupes
CREATE TABLE ftpgroup (
groupname VARCHAR(16) NOT NULL DEFAULT '',
gid SMALLINT (6) NOT NULL default '5500', -- la valeur 5500 est libre de choix
members VARCHAR(16) NOT NULL default '',
PRIMARY KEY (groupname )
) TYPE=MyISAM COMMENT='Groupes pour ProFTP';
-- Table pour les utilisateurs
```

```
CREATE TABLE ftpuser (
id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
userid VARCHAR(32) NOT NULL DEFAULT '',
passwd VARCHAR(32) NOT NULL DEFAULT '',
uid SMALLINT (6) NOT NULL default '5500',
gid SMALLINT (6) NOT NULL default '5500',
email VARCHAR(255) NOT NULL,
homedir VARCHAR(255) NOT NULL DEFAULT '',
shell VARCHAR(16) NOT NULL DEFAULT '/bin/false',
count INT(11) NOT NULL DEFAULT '0',
accessed DATETIME NOT NULL default '0000-00-00 00:00:00',
dl_bytes BIGINT (20) NOT NULL ,
dl_count BIGINT (20) NOT NULL,
ul_bytes BIGINT (20) NOT NULL ,
ul_count BIGINT (20) NOT NULL ,
modified DATETIME NOT NULL default '0000-00-00 00:00:00',
LogInAllowed ENUM('true','false') NOT NULL DEFAULT 'true',
PRIMARY KEY (id)
) TYPE=MyISAM COMMENT='Utilisateurs pour ProFTP';
```

Insertion des données

```
-- insertion d'un compte dans la table pour se connecter
-- ATTENTION ! remplacer le nom et le mot de passe par ceux d'un compte existant sur
-- le serveur Linux si on a choisi RequireValidShell on dans le fichier proftpd.conf
-- Remarque : "Encrypt" utilise le cryptage Unix/Linux, ne fonctionne pas pour MySQL Windows,
-- peut-être que des installations sont possibles pour rendre opérationnel sous Windows (non testé)
INSERT INTO ftpuser
(id ,userid , passwd ,uid ,gid ,homedir,count ,accessed ,modified ,LogInAllowed )
VALUES ('','utilFTP',encrypt('mpUtilFTP'), '5500','5500','/home/utilFTP','','','','true');
-- Ajout d'un groupe dont l'utilisateur est membre
INSERT INTO ftpgroup VALUES ('Groupe FTP','5500','utilFTP');
```

Source internet

- **http://doc.ubuntu-fr.org/proftpd** : quelques explications en français
- **http://www.cgsecurity.org/Articles/proftpd.html** : détail bien expliqué sur la configuration des accès anonymes et sur les droits sur les partages
- **http://www.proftpd.org/docs/directives/linked/configuration.html** : en anglais, sur le site de l'outil, la liste des directives, des modules et leur explication précise
- http://doc.ubuntu-fr.org/proftpd_et_mysql installation *proftpd* et authentification MySQL sur Ubuntu
- Article de Linux Plus Magazine 06/2009

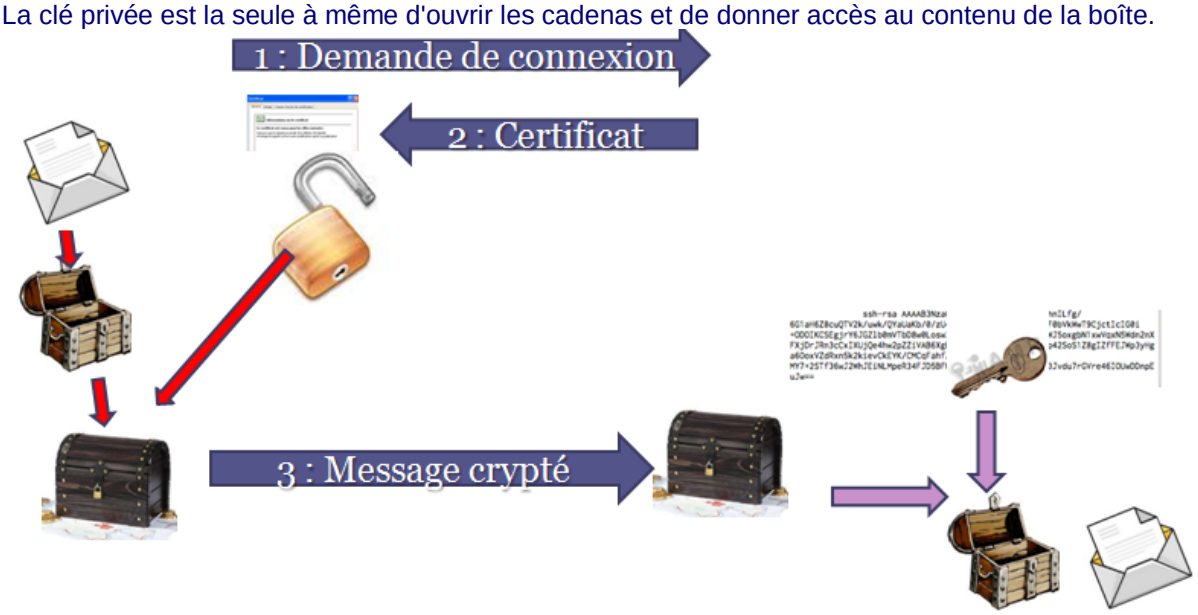
FICHE 6 : CONNEXION ET ECHANGES CRYPTES AVEC SSL/TLS

PRESENTATION

Protection des échanges au-delà du réseau local
La mise en place d'un Firewall permet de protéger le contenu présent à l'intérieur d'un réseau et d'empêcher la fuite de données vers l'extérieur.
Mais pour les échanges autorisés, l'information est diffusée au delà de ***notre sphère de contrôle*** et il faut ajouter une ***capsule de protection*** autour de l'information que l'on transfère (on parle d'***encapsulation***) si l'on souhaite en empêcher la lecture par des **personnes extérieures à l'échange**.

Clé de cryptage
Pour partager des messages de manière confidentielle, deux extrémités d'une communication doivent disposer d'un secret (une ***clé***) qui permettra de coder un message (on parle de ***chiffrement***) à l'émission, et de le décoder (ou encore ***décrypter***) à l'arrivée.
Si c'est la même clé qui permet le cryptage/décryptage, on dira qu'elle est ***symétrique***.
Si on veut un niveau de sécurité accru, on réalisera l'encryptage par le biais d'une clé distribuée à tous les émetteurs potentiels (cette clé est ***publique***), et seul le destinataire possèdera ***la*** clé capable de réaliser le ***déchiffrement*** (la clé est donc ***privée***). On parle alors d'un cryptage ***asymétrique***.

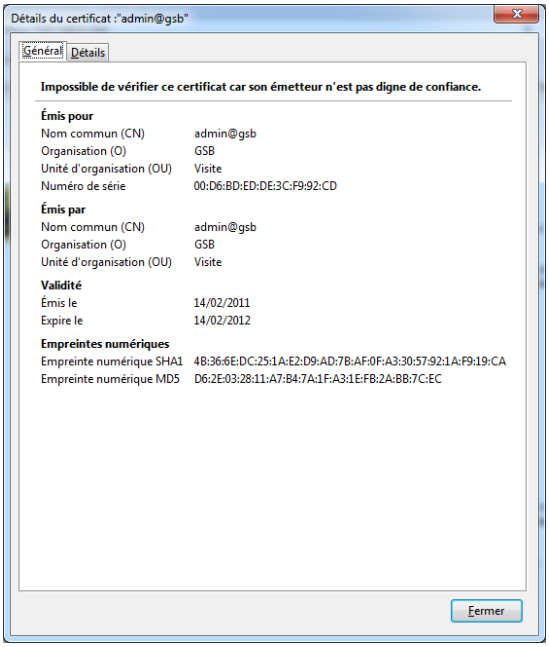
Cryptage, clé publique, clé privée
La clé publique peut être vue comme un cadenas ouvert dont la serrure est basée sur l'empreinte de la clé privée.
On distribue des cadenas (clé publique) à des interlocuteurs qui demandent à entrer en communication..
Ils enferment leur message dans des boîtes qu'ils cadenassent avec cette clé publique (cryptage).
La clé privée est la seule à même d'ouvrir les cadenas et de donner accès au contenu de la boîte.



Certificat
Dans le cas d'un accès à un serveur public sécurisé (par exemple pour gérer ses comptes ou payer en ligne), il est difficile de distribuer la clé publique à l'avance.
Le principe est alors de déposer, sur le serveur, un objet qui remplira les objectifs suivants :

- ***crypter*** l'information envoyée vers le serveur avec la clé publique contenue dans le certificat
- garantir ***l'intégrité*** des données transférées grâce à des techniques de ***hachage***
- garantir l'identité du serveur en précisant toutes les coordonnées de l'entreprise qui le met à disposition et, éventuellement, de l'organisme qui en ***garantit la validité*** (indispensable pour le paiement en ligne)
- indiquer tous les ***algorithmes de cryptage*** supportés par le serveur

Cet élément est un fichier textuel nommé ***certificat***. Divers formats existent, donc le standard de l'Union Internationale des Télécoms (Certificat X.509).



SSL/TLS

Parmi les outils susceptibles de garantir le cryptage, SSL (Secure Socket Layer : Couche de socket sécurisé), défini par la société Netscape en 1994 (V.1) et déployé publiquement en version 2 (1995) s'est imposé comme la technique de sécurisation des échanges en environnement internet.
Il s'applique à tous les protocoles de la couche ***application*** puisqu'il se positionne plus bas dans le modèle OSI (***couche 5 : Session***).
Il supporte les certificats X.509.
TLS (Transport Layer Security) est le nom du protocole depuis 2001, qui remplace la version 3.1 de SSL. Il est maintenant définit par l'IETF (Internet Engineering Task Force) qui se charge de définir les techniques utilisées sur Internet.

La mise en place d'un service sécurisé par TLS permettra d'éviter que l'on puisse exploiter le contenu en cas d'interception des échanges entre le client et le serveur.

PROCEDURE

La démarche est commune quel que soit le service que l'on souhaite sécuriser.
Seuls les fichiers à modifier changent d'un service à l'autre.

1. Créer le répertoire pour stocker les clés et certificats
2. Créer la clé privée
3. Créer le certificat X509 à partir de la clé privée
4. Paramétrer le service concerné
   - activer le module,
   - configurer les fichiers pour qu'ils utilisent la clé et le certificat
   - mettre en écoute sur les ports spécifiques
5. Tester depuis un client en lui indiquant l'adresse et le port adéquats
6. Éventuellement, accepter le certificat auto-signé (**attention**, cela peut être un danger si on ne connaît pas la source).

MODE OPERATOIRE

MISE EN PLACE D'UN CERTIFICAT AVEC OpenSSL

OpenSSL est une transposition OpenSource (**http://www.openssl.org/**) des préconisations de l'IETF pour la mise en place d'une couche sécurisée.
Le nom est resté OpenSSL mais la bibliothèque supporte aussi la version TLS.
OpenSSL gère de nombreux algorithmes de cryptage (AES, DES, RSA,...) et de hachage (SHA, MD5, …).

```
Standard commands
asn1parse       ca              ciphers         crl             crl2pkcs7
dgst            dh              dhparam         dsa             dsaparam
ec              ecparam         enc             engine          errstr
gendh           gendsa          genrsa          nseq            ocsp
passwd          pkcs12          pkcs7           pkcs8           prime
rand            req             rsa             rsautl          s_client
s_server        s_time          sess_id         smime           speed
spkac           verify          version         x509

Message Digest commands (see the `dgst' command for more details)
md2             md4             md5             rmd160          sha
sha1

Cipher commands (see the `enc' command for more details)
aes-128-cbc     aes-128-ecb     aes-192-cbc     aes-192-ecb     aes-256-cbc
aes-256-ecb     base64          bf              bf-cbc          bf-cfb
bf-ecb          bf-ofb          cast            cast-cbc        cast5-cbc
cast5-cfb       cast5-ecb       cast5-ofb       des             des-cbc
des-cfb         des-ecb         des-ede         des-ede-cbc     des-ede-cfb
des-ede-ofb     des-ede3        des-ede3-cbc    des-ede3-cfb    des-ede3-ofb
des-ofb         des3            desx            rc2             rc2-40-cbc
rc2-64-cbc      rc2-cbc         rc2-cfb         rc2-ecb         rc2-ofb
rc4             rc4-40
```

Il propose notamment des commandes pour la création de clé secrète (gendh, genrsa, ...) ou la génération de certificat (x509).

1 : Création d'une clé privée

La base des techniques de cryptage asymétrique repose sur la présence, à un endroit unique, d'une clé privée.
Elle permettra de créer une clé publique (ou un certificat) et de décrypter ce qui aura été chiffré par cette clé publique.

On utilisera (pour l'algorithme RSA) la commande ***genrsa*** de OpenSSL.

openssl genrsa -out *nom_fichier_cle taille_cle*

Exemple : crée une clé privée dans un dossier personnalisé

cd */etc/ssl*
mkdir *mesCles*
cd *mesCles*
openssl genrsa -out *cleGSB.key 1024*

On pourra alors éditer la clé et en lire son contenu.
Cette même clé peut servir à établir des certificats différents pour des usages distincts (serveur FTP, serveur Web, messagerie, etc).

```
-----BEGIN RSA PRIVATE KEY-----
MIICWwIBAAKBgQDX99dnq12oD7iOprE/rt9JwbtX39Ge38oCrwUkUph9NQkVfjKg
cNvvXB8ykpJJHerYuw5ZAiL6r1Yqqu5t9It4EE8OMZYFWioAflnd4CdTcMnMPeSV
bSLJCEKyAZmGhjx0xZyVA75duozC/gMo9eMeCy7Pb2jt8iczpEZUzRJi3QIDAQAB
AoGAK6P2UAmNzFcY6SJ/8Klqj2HpdHFSz3LTREWPmW5jR/G+myw8GLWVi8GmwDZj
/whfm1n5JT1MSjHCDK8rrOZchBBd5p/E55JCAjTg8xSjDcYy2H5rH1ewXXwcn424
hFltNo7SYrwq503gO0ZJv2u9tzV37ndfy4u7S/i5Ez44MJUCQQDyKRmwaFi3XfJr
p8bXRDewhtmqteDFHpb+uLRpvsyRDeGIJetvk0di1BL3RYQcOGlRi+WbDRyUmXLA
YKVCwAXTAkEA5E+JQ0U7czFq5wsXLZMg1IcYXKNLnZBhWanujAq0WlR7xYkJECUQ
9B4xDO0gXZ4LjmjdH4ggPrRc99OXsxUWjwJAL+I+qSpI+KFEEVOQtvpPRMG1gtMK
ChBdbSraydonmzsycHiHEH3/7KPgyDJfyDDDapOOKFW0ZBHfpfi2VPehbwJAFO0x
YKgc3d8etFSw6C1/KdvEIOGa0bNlKOnJgq6Mf2cU/TMlUISFPEr0JgIqIEqXNpNo
CTbZ9cVkbBbJ5dE+AwJAE6CJS5f4SPFKEopvjeJx5IkzDcmcW3MaDnR+VaTORRtT
zlT9yKsDG1bKv2NRB6yhvdAPlTAUqhhSmKlbAgg6GQ==
-----END RSA PRIVATE KEY-----
```

2 : Création d'un certificat X509

La clé privée ne doit jamais être diffusée. On va donc générer la partie ***publique*** sous forme d'un certificat.
Pour que ce dernier soit accessible aux navigateurs, on a recours au format standard X509.

La création va donc passer par deux étapes :

- Création d'un certificat générique (on devra renseigner les coordonnées de l'entreprise)

```
openssl req -new -key nom_cle -out nom_certif_generic.csr
```

- Transformation vers un format X509

```
openssl x509 -req -days nb_jours -in nom_certif_generic.csr -signkey nom_cle -out
nom_certif_X509.crt
```

Exemple : création d'un certificat dans un dossier spécifique à partir de la clé RSA

```
cd /etc/ssl
mkdir mesCertifs
cd mesCertifs
openssl req -new -key ../mesCles/cleGSB.key -out GSBCertGen.csr
openssl x509 -req -days 365 -in GSBCertGen.csr -signkey ../mesCles/cleGSB.key -out
GSBcertif.crt
```

On peut aussi choisir de stocker le certificat directement dans le répertoire du service qu'il concerne (par exemple dans /etc/apache2/mesCertifs)

Le certificat est alors prêt.

Il ne reste plus qu'à configurer les services susceptibles de s'appuyer sur ce certificat.

CONFIGURATION D'APACHE AVEC SSL/TLS

La configuration nécessite l'activation du module SSL pour Apache

```
a2enmod ssl
```

Si le module est déjà activé, un message l'indique.

On peut alors voir dans **/etc/apache2/mods_enabled/ssl.load** que la ligne ci-dessous est dé-commentée

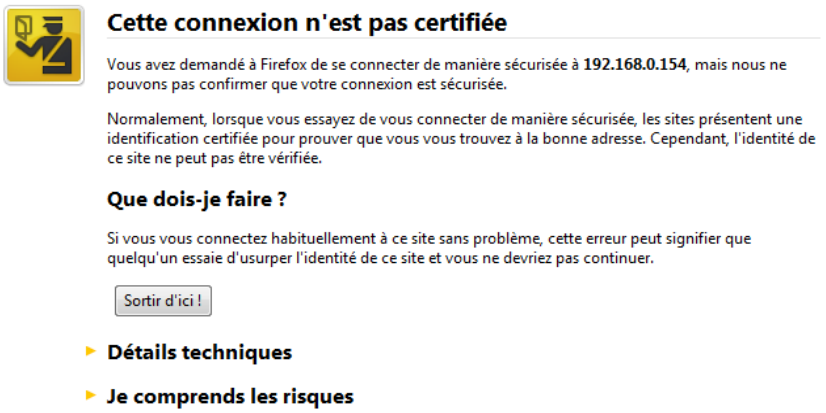
```
LoadModule ssl_module /usr/lib/apache2/modules/mod_ssl.so
```

Dans le fichier **ports.conf** de Apache, on vérifiera que le serveur écoute sur le port standard 443 (ou sur un autre port si on souhaite faire une configuration personnalisée).
Dans le fichier **/etc/apache2/sites-enabled/000-Default** (ou dans **httpd.conf**), on ajoutera un hôte virtuel pour cette écoute :

```
#on adaptera le numéro de port conformément à ce qui a été écrit dans ports.conf
<VirtualHost *:443>
        DocumentRoot /var/www
        SSLEngine on ; active le SSL
        SSLCertificateFile /etc/ssl/mesCertifs/GSBCertif.crt ; chemin du certificat X509
        SSLCertificateKeyFile /etc/ssl/mesCles/cleGSB.key ; chemin de la clé privée
</VirtualHost>
```

Il faudra redémarrer Apache, qui indiquera si une erreur éventuelle est rencontrée (dans le chemin, dans le nom du fichier, dans le contenu du certificat, etc).

Contrôle depuis un navigateur
Dans la barre de navigation du navigateur, on tapera https://adresseServeur.



Du fait que le certificat que nous avons créé n'est pas garanti par un organisme officiel, le navigateur mettre en garde sur le manque de confiance accordée à un certificat signée par son créateur.
On doit vérifier qu'un cadenas fermé au bas du navigateur atteste d'une navigation sécurisée.

CONFIGURATION DE ProFTP AVEC SSL/TLS

Pour commencer, on devra indiquer dans le fichier ***/etc/proftpd/modules.conf*** qu'il faut activer TLS

```
LoadModule mod_tls.c
```

On ira ensuite préciser au fichier ***proftpd.conf*** d'inclure le fichier de configuration de tls en dé-commentant la ligne suivante :

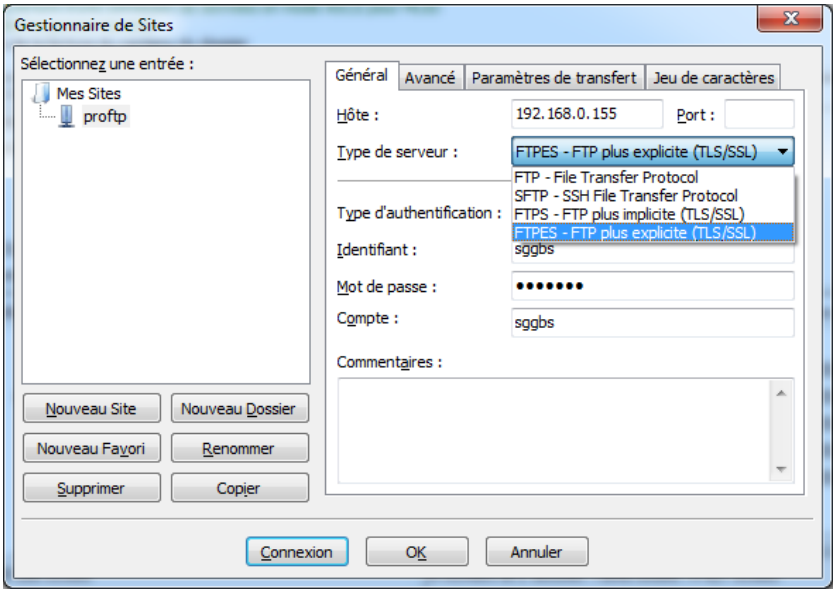
```
Include /etc/proftpd/tls.conf
```

Dans ce fichier ***tls.conf*** on devra trouver au minimum les éléments suivants :

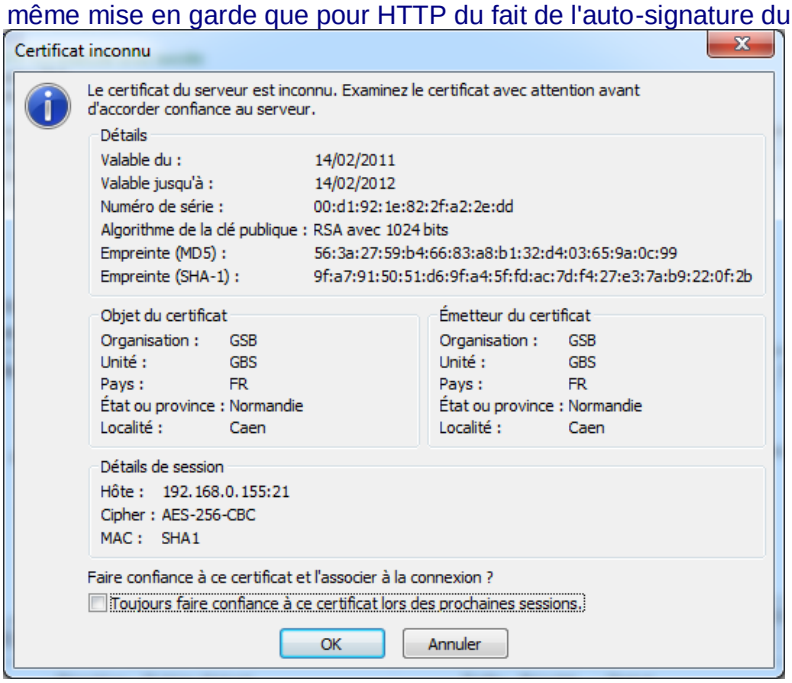
```
<IfModule mod_tls.c>
        TLSEngine on ;active le TLS
        TLSLog /var/log/proftpd/tls.log ; dossier pour enregistrer les journaux tls
        TLSProtocol SSLv23 ; versions supportées (2 et 3)

        TLSRSACertificateFile /etc/ssl/mesCertifs/GSBCertif.crt ;chemin du certif
        TLSRSACertificateKeyFile /etc/ssl/mesCles/cleGSB.key ; chemin de la clé

        TLSVerifyClient off ; n'oblige pas l'authentification des clients pour TLS
        #TLSRequired on ; peut obliger les clients à utiliser TLS

</IfModule>
```

Il faudra bien entendu redémarrer le service ***proftpd*** qui indiquera si une erreur est rencontrée (dans le chemin, dans le nom du fichier, dans le contenu du certificat, etc).

Test depuis le client
On accèdera depuis un client FTP en contactant le serveur par une connexion FTP SSL/TLS Explicite (ici sous FileZilla Client).



On rencontrera la même mise en garde que pour HTTP du fait de l'auto-signature du certificat.



Sources

- **http://httpd.apache.org/docs/trunk/fr/ssl/ssl_intro.html** : sur le site de ***Apache***, une explication détaillée des principes du chiffrement, du rôle d'un certificat, des déclinaisons de SSL/TLS
- http://www.vanemery.com/Linux/Apache/apache-SSL.html : un tutoriel (en anglais) assez complet qui présente les diverses manipulations pour sécuriser un serveur web Apache, avec authentification par certificat côté client, demande de mot de passe pour se connecter au site, etc.