



TP 2 : Contrôle de l'accessibilité

Contrôle des accès à un répertoire

- Chaque répertoire dont le contenu doit être géré par Apache peut être configuré en particulier. (ceci s'applique aussi à ses sous-répertoires) Le paramétrage du *répertoire* est précisé par un ensemble de clauses placées entre les balises `<Directory chemin_répertoire>` et `</Directory>` que l'on trouvera dans le fichier de configuration du site.
- Attention : contrairement aux permissions Unix, les clauses s'appliquent AUSSI à TOUS les sous-répertoires, SAUF s'il existe une directive du genre `<Directory sous-rep>` qui s'applique spécifiquement à l'un des ses sous-répertoires *sous-rep*. Dans ce cas, les nouvelles directives remplacent le paramétrage du répertoire parent.
On peut utiliser aussi `<Location >`, semblable à `Directory`, mais en spécifiant une URL, plutôt qu'un chemin de répertoire.

- Exemple :

```
#----- Global Access Configuration -----  
# Voici un exemple de paramétrage des permissions d'accès. Il est préférable  
# d'être restrictif à la racine et d'étendre les permissions à des répertoires spéciaux.  
  
# Politique par défaut : accès interdit à tous à partir de /  
# (donc pour tout le système de fichier) sauf permissions à donner explicitement  
<Directory />  
order deny, allow  
deny from all  
Options None  
AllowOverride None  
</Directory>  
  
# La racine du serveur WEB, il faut bien en permettre l'accès  
<Directory /var/www/>  
AllowOverride None  
order allow,deny  
# allow from = all pour permettre au monde entier d'accéder aux documents
```

```
allow from all
</Directory>
```

```
# pour le répertoire contenant les scripts cgi
<Directory /var/www/cgi-bin>
AllowOverride None
Options ExecCGI
</Directory>
```

- **Règles à appliquer pour restreindre les accès**

Pour un répertoire donné, dans son conteneur <Directory>, on peut préciser la liste des hôtes (le séparateur est l'espace) dont les requêtes seront traitées, et ceux dont les requêtes seront rejetées.

On précise d'abord une règle générale avec la directive **order allow, deny** ou l'inverse, qui précise la règle principale à appliquer aux machines qui figurent sur les listes explicites qui suivent les clauses **allow from** et **deny from**

- order allow, deny : autorise les hôtes de la liste allow, mais rejette ceux de la liste deny
- order deny, allow : rejette les hôtes de la liste deny, mais autorise ceux de la liste allow
- exemple1 : soit à autoriser tout le réseau 172.16.0. **sauf** 172.16.0.25

```
# quel est le bon ordre : order deny, allow ou bien allow, deny ?
order ....
```

```
# par contre l'ordre relatif de ces 2 clauses suivantes n'a pas d'importance
allow from 172.16.0.0/255.255.255.0
deny from 172.16.0.25
```

- Modifier l'accès à votre site par défaut pour autoriser l'accès à toutes les machines du réseau **sauf** celles qui vous sont voisines. Bien vérifier.

- Interdire l'accès de tous les hôtes sauf localhost, et deux autres machines. Vérifier.

- Les paramètres d'**Options** permettent de contrôler l'action d'Apache sur les répertoires

Option	signification
All None	toutes aucune option(s) permise(s)
ExecCGI	exécution de scripts autorisée
FollowSymLinks	le serveur suivra les liens symboliques rencontrés

	dans le répertoire
Indexes	autorise l'affichage du listing d'un répertoire (si un fichier par défaut n'y est pas trouvé : index.html ...)

Rechercher les lignes comportant le mot-clé **indexes** dans le fichier de configuration de votre site par défaut, changer le comportement par défaut, en permettant ou en inhibant l'effet de **indexes** sur le répertoire racine de votre site. Bien tester ce mécanisme.

Autoriser les pages WEB personnelles

Cette manipulation va permettre aux utilisateurs de publier leurs "pages persos" sur un serveur WEB sans être administrateur ou sans avoir les droits particuliers.

- Tout naturellement, ces documents devront être placés dans leur répertoire personnel dont ils sont propriétaires et ont le plein accès, tout en permettant l'accès et la lecture à tous.
- La page d'accueil doit être présente dans un sous répertoire du répertoire perso. Par défaut le nom de ce sous répertoire est fixé à **public_html**
- Ce sous répertoire des pages personnelles a un nom spécifié par la directive **UserDir** gérée par le module apache UserDir . Donc par défaut, on y trouve : **UserDir public_html**
- Plus concrètement, si la page d'accueil s'appelle *index.html*, pour l'utilisateur *toto* , sur le serveur, son ouverture sur une station du réseau est obtenue en spécifiant l'URL :
http://serveur/~toto/index.html.

Attention aux droits d'accès !

L'accès à ces pages persos exige que le droit de parcours **x** soit accordé à tous les rép. des utilisateurs et à **public_html**(droits 771), et que le droit de lecture **r** soit bien sûr accordé aux fichiers des pages html (droits 644)

- Activer s'il le faut le module UserDir
- Vérifier sa configuration
- Créer dans votre répertoire personnel le répertoire nécessaire à l'hébergement

des pages web personnelles et y créer un fichier html.

- Vérifier que les droits d'accès soit suffisants pour qu'apache puisse accéder à votre fichier html.
- Tester l'accès à votre fichier html avec un navigateur en formatant correctement l'URL