VII

Sécurité sous Apache

Exécution des processus

User et Group

Ces directives définissent les EUID et EGID des processus-fils d'Apache.
On privilégiera un utilisateur et un groupe non privilégiés pour circonscrire l'impact d'une faille de sécurité sur le système. Il est également possible de lancer Apache en environnement "chrooté", au détriment de la facilité d'administration.
Le processus-père appartient toujours à root.

Exemple (par défaut):

```
User            apache
Group           apache
```

A l'exécution, une visualisation des processus donne:

```
root     root     httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
apache   apache   \_ httpd
```

Pour des raisons de sécurité, les processus-fils ne peuvent appartenir au super-utilisateur (root).

Portée des balises

Il est possible de masquer, limiter ou empêcher l'accès à certains répertoires, fichiers ou URLs aux clients ainsi qu'à certaines portions du site, par voie de conséquence.

<Directory>

Les directives contenues dans ce bloc portent sur le répertoire positionné en argument et ses sous-répertoires. La balise `<Directory />` configure l'ensemble des documents de tous les sites.

Exemple:

```
<Directory       "/var/www/html/mon_site/images">
        directives
</Directory>
```

<Files>

Les directives contenues dans ce bloc portent sur le fichier positionné en argument.

Exemple:

```
<Files           "/var/www/html/mon_site/images/mon_image.png">
        directives
</Files>
```

<Location>

Les directives contenues dans ce bloc portent sur l'URL relative positionnée en argument.

Exemple:

```
<Location        /icons/index.html>
        directives
</Location>
```

Expressions rationnelles

~

La portée des balises précédentes peut être augmentée par l'utilisation de caractères génériques dans des expressions rationnelles. Elles sont déclarées par le caractère "~" (tilde) entre la balise etl'argument.

Configuration d'un ensemble de répertoires:

```
<Directory ~ "/var/www/html/do*">
        directives
</Directory>
```

Configuration d'un ensemble de fichiers:

```
<Files ~ "*.gif">
        directives
</Files>
```

Configuration d'un ensemble d'URLs relatives:

```
<Location ~ /icones/*.png>
        directives
</Location>
```

Octroi de capacités

Options

Dans le bloc <Directory>, cette directive autorise certaines capacités du serveur Apache pour le répertoire considéré.

Exemple:

```
<Directory "/var/www/html/docs">
        Options All MultiViews                  # Active toutes les capacités ('All' seul n'est pas suffisant).
</Directory>
```

Liste exhaustive des options possibles:

- All: toutes les capacités activées sauf `MultiViews`,
- None: aucune des capacités qui suivent,
- FollowSymLinks: Apache suivra les liens symboliques (même en dehors du répertoire),
- SymLinksIfOwnerMatch le lien et sa cible doivent avoir le même propriétaire pour être autorisés,
- Indexes: autorise l'indexation de contenu (listing de répertoire),
- Includes: autorise le langage de script Apache (les SSI: ServerSidesIncludes),
- IncludesNoExec:: active les SSI sauf la commande `exec`,
- ExecCGI: active l'exécution par le système de scripts CGI,
- MultiViews: doit être précisé même si `Options All`. Si le client demande une page /le/chemin/liens, qui n'est pas présente dans le répertoire, avec l'option `MultiViews` le serveur renverra la page /le/chemin/liens.* présente dans l'arborescence.

L'utilisation de liens doit être faite avec précaution, en utilisant de préférence des chemins relatifs,...

Restriction d'accès réseau

Allow from

Autorise l'accès à partir d'un nom de domaine ou d'une adresse IPv4 ou IPv6, qu'ils soient spécifiées de manière totale ou partielle.

Pour autoriser un réseau, on précisera le masque sous la forme décimale pointée ou CIDR.

Exemple:

```
<Directory "/var/www/html/intranet">
        Allow from 192.168.0.0/24 10.23 .net ::1
        Order deny,allow
</Directory>
```

Deny from

Restreint l'accès des clients selon le nom de domaine ou l'adresse IP, spécifiées de manière totale ou partielle.

Pour restreindre l'accès d'un réseau, on précisera le masque sous la forme décimale pointée ou CIDR.

Exemple:

```
<Directory "/var/www/html/intranet">
        Deny from 82.233.232/24 .com 10.23.0.0/16
        Order allow,deny
</Directory>
```

Restriction d'accès réseau

Order

Dans le cas où les directives `Allow from` et `Deny from` sont renseignées, fixe l'ordre d'application entre autorisations et permissions.

Si plusieurs directives Allow from et `Deny from` sont présentes, elles s'appliquent séquentiellement.

Exemple:

```
<Directory "/var/www/html/intranet">
        Allow from 192.168.0.0/24
        Deny from All
        Order deny,allow                    # deny,allow et allow,deny sont des mots-clés d'un seul tenant.
</Directory>
```

SetEnvIf

Positionne une variable d'environnement selon une composante de larequête.

Exemple de la documentation officielle:

```
SetEnvIf User-Agent ^KnockKnock/2\.0 let_me_in
<Directory /docroot>
        Order Deny,Allow
        Deny from all                       # Seuls les clients web dont l'identifiant commence par
        Allow from env=let_me_in            # 'KnockKnock/2.0' sont autorisés.
</Directory>
```

Restriction d'accès par méthode

<Limit>

La balise `<Limit>` restreint la portée des directives aux méthodes HTTP spécifiées en argument.

Le contrôle d'accès (i.e. les directives) est inopérant sur les autres méthodes.

Exemple:

```
<Directory "/var/www/html/formulaires">
        <Limit POST PUT DELETE>                 # L'envoi et la suppression de données sur le serveur
                require valid-user              # requérra un utilisateur authentifié.
        </Limit>
</Directory>
```

<LimitExcept>

La balise `<LimitExcept>` restreint la portée des directives aux méthodes HTTP non spécifiées en argument.

La balise `<LimitExcept>` agit également sur les méthodes inconnues (tentative de corruption du serveur) et est donc préférable à la balise `<Limit>`.

Exemple:

```
<Directory "/var/www/html/formulaires">
        <LimitExcept GET POST>                  # L'accès réseau est interdit à tous pour les méthodes
                Deny from all                   # autres que GET et POST.
                Order allow,deny
        </LimitExcept>
</Directory>
```

- Les **méthodes** HTTP sont **sensibles à la casse**.

`HEAD` est équivalente à `GET`. `TRACE` n'est pas prise en compte.