Sécuriser un serveur Apache

Serveur Linux CentOS

Table des matières

1 Introduction

Le service **apache** est un service web très populaire, performant, et sa conception modulaire le dote d'une grande richesse fonctionnelle. Découvrez comment le sécuriser.

2 Afficher le moins d'informations

Il est très facile de découvrir quel serveur tourne sur un site web comme le montre l'exemple suivant :

```
[root@localhost ~]# telnet 127.0.0.1 http
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
HEAD / HTTP/1.0
```

```
HTTP/1.1 200 OK
Date: Thu, 22 Jun 2017 10:40:13 GMT
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
Last-Modified: Thu, 22 Jun 2017 09:57:50 GMT
ETag: "5-552898482bcbf"
Accept-Ranges: bytes
Content-Length: 5
Connection: close
Content-Type: text/html; charset=UTF-8

Connection closed by foreign host.
```

Si aucune page d'accueil n'est définie dans /var/www/html, vous recevrez l'erreur suivante :

```
[root@localhost ~]# telnet 127.0.0.1 http
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
HEAD / HTTP/1.0
```

```
HTTP/1.1 403 Forbidden
Date: Thu, 22 Jun 2017 10:57:40 GMT
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
Last-Modified: Thu, 16 Oct 2014 13:20:58 GMT
ETag: "1321-5058a1e728280"
Accept-Ranges: bytes
Content-Length: 4897
Connection: close
```

Un pirate apprend que le service **apache** tourne sous une distribution **CentOS.**

On limite la divulgation d'information en insérant dans le fichier de configuration /etc/httpd/conf/httpd.conf**,** la ligne :

```
ServerTokens Prod
```

```
ServerTokens Prod[uctOnly]
        Server: Apache
ServerTokens Major
        Server: Apache/2
ServerTokens Minor
        Server: Apache/2.4
ServerTokens Min[imal]
        Server: Apache/2.4.6
ServerTokens OS
        Server: Apache/2.4.6 (CentOS)
ServerTokens Full
        Server: Apache/2.4.6 (CentOS) PHP/5.4.16
```

```
# systemctl restart httpd
```

```
[root@localhost ~]# telnet 127.0.0.1 http
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
HEAD / HTTP/1.0
```

```
HTTP/1.1 403 Forbidden
Date: Thu, 22 Jun 2017 11:00:39 GMT
Server: Apache
Last-Modified: Thu, 16 Oct 2014 13:20:58 GMT
ETag: "1321-5058a1e728280"
Accept-Ranges: bytes
Content-Length: 4897
Connection: close
Content-Type: text/html; charset=UTF-8

Connection closed by foreign host.
```

Ainsi, la bannière :

```
Server: Apache/2.4.6 (CentOS) PHP/5.4.16
```

se limite à la bannière :

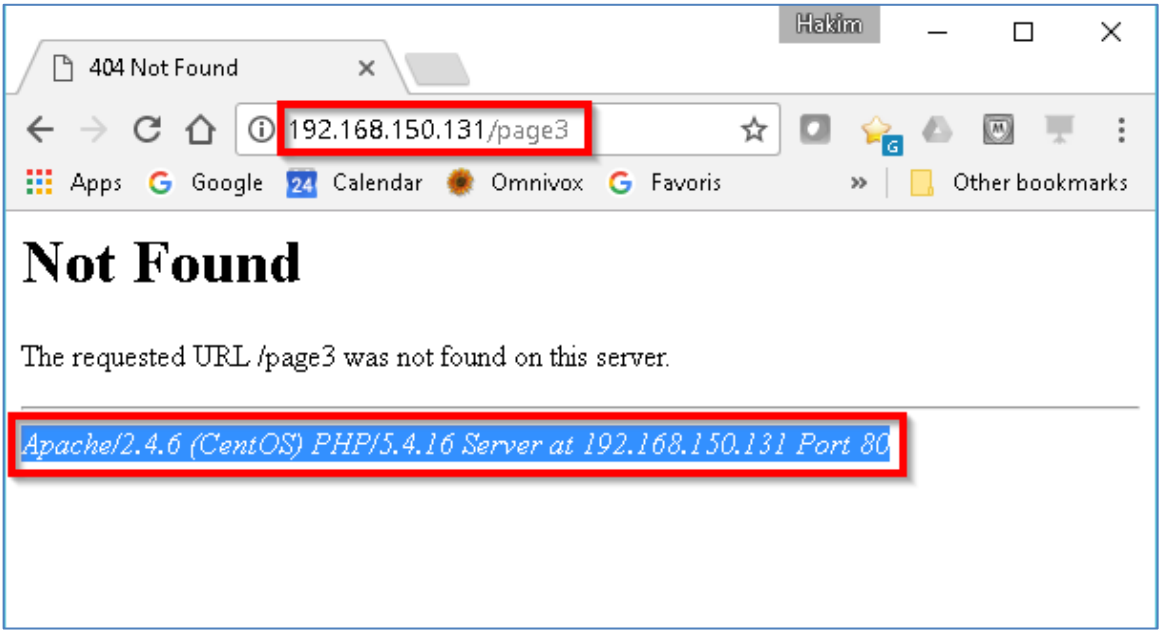
```
Server: Apache
```

Sur CentOS, le défaut est **ServerTokens OS**

3 Désactiver la signature

Cela ne suffit toujours pas à masquer la version du service **apache**. Si la signature est activée (ServerSignature On).

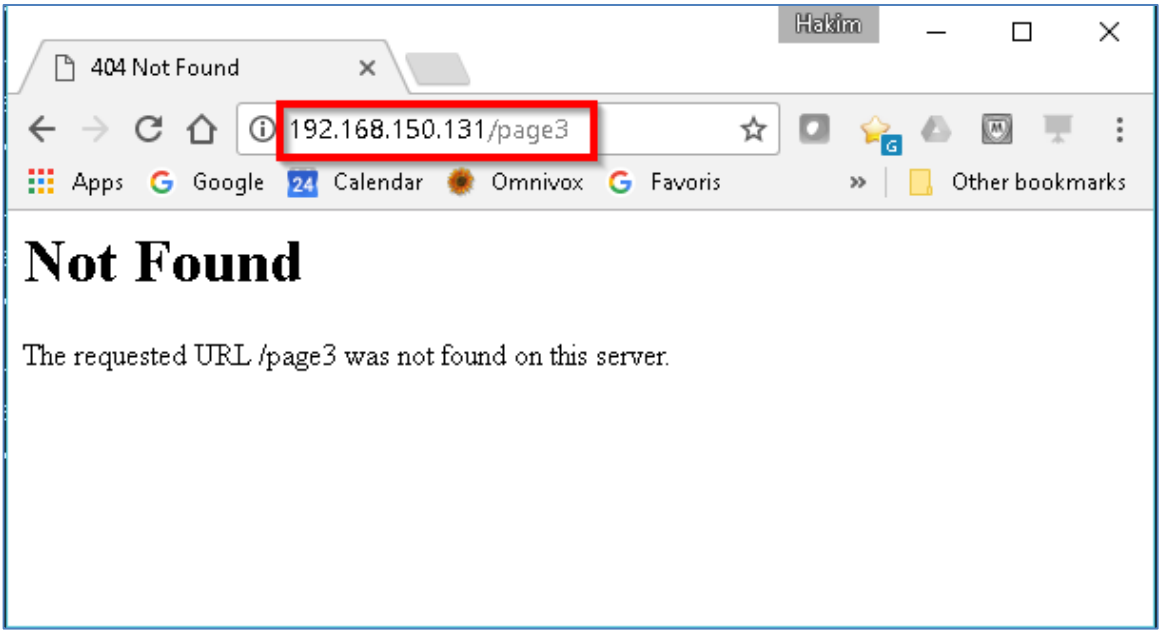
Si vous demandez une page inexistante, **apache** renvoie une page d'erreur **404** avec en bas de la page, le message *Apache/2.4.6 (CentOS) PHP/5.4.16 Server at 192.168.150.131 Port 80* qui révèle la version du service **apache**.



Pour empêcher cela, il faut désactiver l'insertion de la signature du serveur avec la commande :

```
ServerSignature Off
```

```
# systemctl restart httpd
```



Il est préférable d'utiliser **ErrorDocument 404 /missing.html** pour définir votre propre page d'erreur **404**.

4 Limitations des attaques

De façon à limiter la portée des attaques de type **DoS** (Denial of Service), il est conseillé de limiter le nombre de connexions simultanées **MaxClients** et en particulier le nombre de connexions persistantes **MaxKeepAliveRequests**.

Les connexions persistantes permettent d'effectuer des requêtes successives lors de la même connexion, ce qui augmente les performances du serveur.

L'utilisation d'un timeout empêche les connexions sans fin.

```
MaxClients 150
MaxKeepAliveRequests 100
KeepAliveTimeout 5
```

```
# systemctl restart httpd
```

5 Bien définir un virtual host

apache permet la définition de serveurs virtuels, c'est-à-dire que le même serveur peut héberger, y compris sur une même adresse IP, plusieurs sites différenciés par leur nom. Pour limiter les risques liés à une panne des serveurs **DNS** ou à des manipulations frauduleuses, il convient de définir le **VirtualHost** par une adresse IP puis de préciser son nom.

```
<VirtualHost 194.57.201.103>
ServerName www.orabec.ca
</VirtualHost>
```

6 Gérer ses fichiers de log

apache permet de définir ses propres formats **LogFormat** pour les enregistrements dans les fichiers de log.

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"
\"%{User-Agent}i\"" combined
```

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
```

```
LogFormat "%{Referer}i -> %U" referer
```

```
LogFormat "%{User-agent}i" agent
```

Format	Description
%a	Remote IP-address
%A	Local IP-address
%B	Size of response in bytes, excluding HTTP headers.
%b	Size of response in bytes, excluding HTTP headers. In CLF format, *i.e.* a '–' rather than a 0 when no bytes are sent.
%D	The time taken to serve the request, in microseconds.
%h	Remote host
%H	The request protocol
%{*Foo*}i	The contents of *Foo*: header line(s) in the request sent to the server.
%l	Remote logname (from identd, if supplied). This will return a dash unless IdentityCheck is set On.
%p	The canonical port of the server serving the request
%P	The process ID of the child that serviced the request.
%q	The query string (prepended with a ? if a query string exists, otherwise an empty string)
%r	First line of request
%s	Status. For requests that got internally redirected, this is the status of the *original* request --- %...>s for the last.
%t	Time the request was received (standard english format)
%T	The time taken to serve the request, in seconds.
%u	Remote user (from auth; may be bogus if return status (%s) is 401)
%U	The URL path requested, not including any query string.
%v	The canonical ServerName of the server serving the request.

Ensuite, on enregistre les informations de log précisées par le format dans le fichier de son choix :

```
CustomLog /var/log/httpd/access_log common
```

Suivant l'utilisation de ces fichiers de logs (reporting,...), il peut être intéressant de faire apparaître le nom des machines se connectant au serveur web au lieu de leur adresse IP : **HostNameLookups** On active la résolution inverse.

7 Gestion des droits

Nous présenterons ici les mesures préventives liées aux fichiers contenus dans l'arborescence du serveur web.

7.1 Directory, Files, Location

La gestion des accès est effectuée par le module **mod_access**. On manipule principalement trois catégories d'objets :

- **Directory** désigne un répertoire du serveur
- **Location** une arborescence du serveur web
- **Files** un fichier

Voici un exemple :

```
<Directory /docroot>
  order deny,allow
  deny from all
  allow from www.orabec.ca
</Directory>
```

Il est fortement conseillé de tout interdire par défaut :

```
<Directory />
  Order deny,allow
  Deny from all
</Directory>
```

Ensuite, il ne reste qu'à valider l'accès aux répertoires correspondant aux sites

Order indique dans quel ordre les directives deny et allow sont évaluées.

Deny from all interdit l'accès depuis partout. On aurait pu indiquer un nom de machine, un nom de domaine, une adresse IP, un couple IP/masque de réseau.

7.2 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 de répertoires est précisé par un ensemble de clauses placées entre les balises :

<Directory repertoire> et **</Directory>**

Contrairement aux permissions Unix, les clauses s'appliquent aussi à tous les sous-répertoires. Sauf s'il existe une directive du genre :

<Directory sous-repertoire>

qui s'applique spécifiquement à l'un de ses sous-répertoires. Dans ce cas, les nouvelles directives supplantent 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

```
<Directory />
order deny, allow
deny from all
Options None
AllowOverride None
</Directory>
```

```
<Directory /var/www>
  Options Indexes Includes FollowSymLinks
  allow from all
</Directory>
```

```
<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 ou 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**.

Si **order** n'est pas spécifié alors l'ordre est : **deny, allow**

Client	Order Allow,Deny	Order Deny,Allow
Match Allow seulement	Requête permise	Requête permise
Match Deny seulement	Requête non permise	Requête non permise
Pas de match	Requête non permise	Requête permise
Match Allow et Deny	Requête non permise	Requête permise

EXEMPLE

Autoriser tout le réseau **172.16.0.** sauf **172.16.0.25**

Quel est le bon ordre : **deny, allow** ou bien **allow, deny** ?

```
Order allow,deny
allow from 172.16.0.0/255.255.255.0
deny from 172.16.0.25
```

7.3 Options

La directive **Options** permet de contrôler les fonctionnalités du service **apache** sur un répertoire particulier.

Elle peut prendre l'une des valeurs suivantes :

None

Aucune option.

All

Regroupe les différentes options sauf **MultiViews**. C'est la valeur par défaut.

ExecCGI

Permettre l'exécution des scripts **CGI** en utilisant le module **mod_cgi**.

FollowSymLinks

Suivre les liens symboliques dans le répertoire.

Includes

Permettre les Server-side includes en utilisant le module mod_include.

IncludesNOEXEC

Les Server-side includes sont permis, mais les commandes **#exec cmd** et **#exec cgi** sont désactivées.

Indexes

Si on tente d'accéder à un URL correspondant à un répertoire et qu'il n'existe aucun fichier dans **DirectoryIndex** (exemple **index.html**) alors la liste formatée du répertoire est retournée (**mod_autoindex**).

MultiViews

Redirige une demande pour **index.html** vers **index.html.en** ou **index.html.fr** selon la préférence signalée par le navigateur au serveur web (mod_negotiation).

SymLinksIfOwnerMatch

Le serveur suit les liens symboliques des cibles qui ont le même propriétaire que les liens.

Il est important d'être le plus restrictif possible par défaut, il est conseillé de n'autoriser que le suivi des liens symboliques où liens et destinations ont le même propriétaire :

```
<Directory />
    Options SymLinksIfOwnerMatch
    AllowOverride None
</Directory />
```

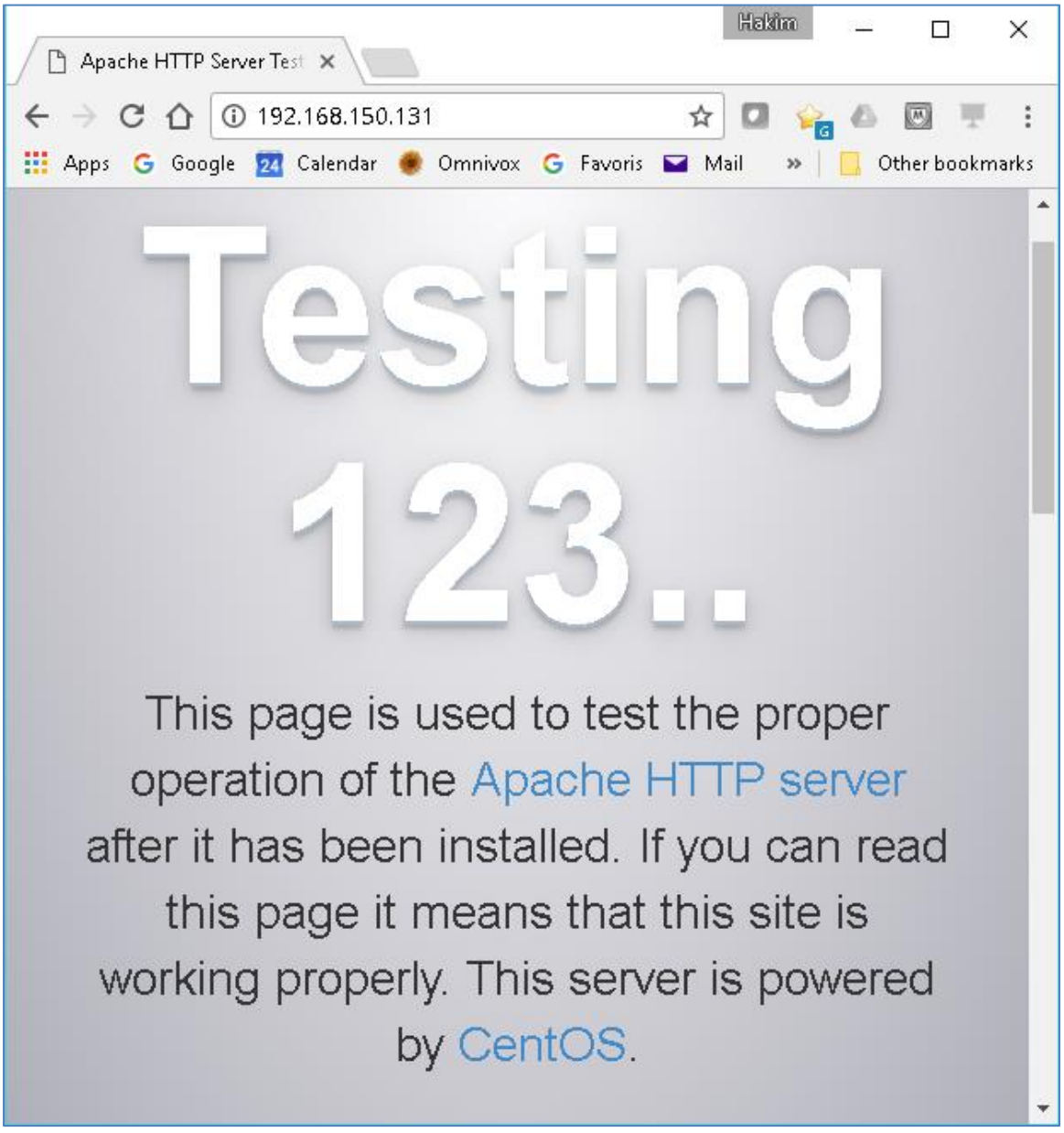
7.4 Indexer un répertoire

Utiliser l'option **Indexes**.

L'URL http://localhost permet d'afficher la page d'accueil se trouvant dans **DocumentRoot** (**/var/www/html**).

Cette page d'accueil doit se trouver dans **DirectoryIndex** sinon c'est la page de bienvenue par défaut qui est affichée:

/var/www/error/noindex.html



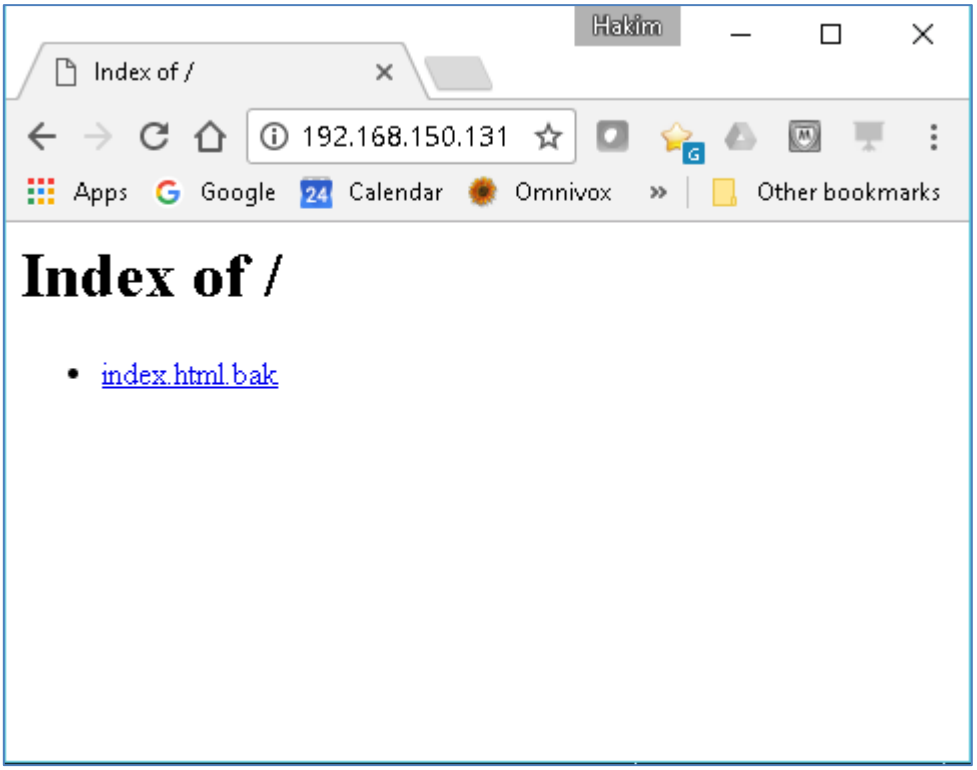
Tel que spécifier dans le fichier :

/etc/httpd/conf.d/welcome.conf

Pour désactiver la page de bienvenue, il suffit de commenter la ligne suivante dans le fichier de configuration d'Apache :

IncludeOptional conf.d/.conf*

Vous pouvez aussi renommer le fichier : **/etc/httpd/conf.d/welcome.conf**



7.5 AllowOverride

Cette directive contrôle le dispositif **.htaccess**.

8 Le dispositif .htaccess

- Principe

  apache permet de délocaliser la gestion de la configuration, au moyen de fichiers spéciaux appelés par défaut **.htaccess**.

  Chaque fichier .**htaccess** est placé directement dans le répertoire dont il doit gérer la configuration particulière et éventuellement protéger l'accès.

  Ce mécanisme de délégation doit au préalable être soumis à autorisation comme décrit ci-dessous.

- Syntaxe

  La clause **AccessFileName .htaccess** fixe le nom du fichier dont la présence dans un répertoire est considérée comme une directive de configuration pour ce répertoire.

  La syntaxe à employer dans ces fichiers **.htaccess** est identique à la syntaxe utilisée dans **httpd.conf**. On peut en particulier y inclure des restrictions d'accessibilité par hôte et des autorisations d'accès par utilisateur.

- Fonctionnement

  Les fichiers **.htaccess** étant lus dynamiquement au moment de chaque requête qui concerne son répertoire, toute modification de ces fichiers prend effet immédiatement, contrairement à **httpd.conf** pour lequel il est nécessaire de faire relire la configuration au serveur.

  Mais alors n'y aurait-il pas possibilité de conflit avec les directives placées dans **httpd.conf** dans un conteneur de directives **<Directory chemin-rép> ...</Directory>** ?

  C'est le rôle de la directive **AllowOverride** de préciser la manière selon laquelle les directives contenues dans un fichier **.htacces** doivent être prises en compte, si ces directives ont "le droit" de supplanter ou non celles qui sont incluses dans la présente directive.

  Ainsi, l'administrateur a le dernier mot ! S'il veut inhiber totalement l'action de **.htaccess**, il précisera **AllowOverride NONE** pour le répertoire. Sinon, il peut accorder des droits complets au fichier **.htaccess** avec **ALL** (prise en compte totale) ou limités en ne positionnant que certaines valeurs. On limite souvent cette délégation de gestion à **AllowOverride AuthConfig** ou **AuthUserFile**, ce qui est suffisant pour protéger l'accès à un site privé par une authentification.

9 Page Web des utilisateurs UserDir

Les utilisateurs du serveur peuvent bien souvent publier leurs pages dans leur répertoire **public_html**, configuré via le paramètre **UserDir public_html**. De façon à éviter une mauvaise surprise, root n'est pas autorisé à faire de même :

```
UserDir disabled root
```

10 User et Group

Le serveur web est lancé par l'utilisateur **root** ce qui lui permet d'utiliser le port privilégié **80**, ensuite il prend l'identité d'un utilisateur sans pouvoir **apache**.

```
User apache
Group apache
```

11 Protection d'une page

11.1 Protection par usager

Le module **mod_auth** permet de protéger l'accès à un répertoire par mot de passe. En pratique, c'est souvent utilisé pour filtrer les accès à un répertoire d'une page personnelle.

La protection d'une page se fait de manière très simple, tous les fichiers à accès limité devraient être concentrés dans un même répertoire. Par exemple **/var/www/html/secret**.

Méthode 1

Étape 1

```
# mkdir /var/www/html/secret
```

Étape 2

Il ne faut pas oublier de créer un fichier index.html dans ce répertoire pour pouvoir faire un test.

```
# vi /var/www/html/secret/index.html
Page secrète
```

Étape 3

Créer le fichier **hpasswd** et ajouter un usager :

```
# htpasswd -c /etc/httpd/conf/hpasswd hakimb
New password:
Re-type new password:
Adding password for user hakimb
```

L'option **–c** permet de créer le fichier **hpasswd**. S'il existe déjà alors il sera écrasé.

Pour ajouter un usager sans écraser le fichier **hpasswd** :

```
# htpasswd  /etc/httpd/conf/hpasswd usager88
```

Étape 4

Modifier le fichier de configuration httpd.conf :

```
# vi /etc/httpd/conf/httpd.conf
<Directory "/var/www/html/secret">
  AuthType Basic
  AuthName "Zone protege"
  AuthUserFile /etc/httpd/conf/hpasswd
   require valid-user
</Directory>
```

Étape 5

```
# service httpd restart
Stopping httpd:                                            [  OK  ]
Starting httpd:                                            [  OK  ]
```

Étape 6

Tester l'URL **http//localhost/secret**

Méthode 2

Une autre méthode consiste à créer un fichier nommé **.htaccess** dans le répertoire **/var/www/html/secret** :

Étape 1

```
# vi /var/www/html/secret/.htaccess
  AuthType Basic
  AuthName "Zone protege"
  AuthUserFile /etc/httpd/conf/hpasswd
  <Limit GET >
   require valid-user
  </Limit>
```

Étape 2

Par défaut, l'utilisation des fichiers **.htaccess** est désactivée. Il faudrait donc l'activer pour ce répertoire :

```
vi /etc/httpd/conf/httpd.conf
```

```
<Directory "/var/www/html/secret">
#    AllowOverride None
#    AllowOverride All
     AllowOverride AuthConfig
</Directory>
```

A noter que le fichier **.htaccess** peut être nommé différemment en utilisant la directive **AccessFileName**

Étape 3

Redémarrer le service **apache** :

```
# service httpd restart
Stopping httpd:                                [  OK  ]
Starting httpd:                                [  OK  ]
```

Étape 4

Tester l'**URL http//localhost/secret**

Par précaution, il faut empêcher un utilisateur de voir le contenu du fichier **.htaccess** récupérer via le web (par défaut existe déjà dans le fichier **httpd.conf**):

```
AccessFileName .htaccess

<Files ~ "^\.ht">
    Order deny,allow
    Deny from all
</Files>
```

11.2 Protection par groupe

Étape 1

```
# mkdir /var/www/html/achat
```

Étape 2

Il ne faut pas oublier de créer un fichier **index.html** dans ce répertoire pour pouvoir faire un test.

```
# vi /var/www/html/achat/index.html
Page Achat
```

Étape 3

Créer le fichier **hgroup** (groupes):

```
# vi /etc/httpd/conf/hgroup
technique:tony
marketing:amelie anne
```

Étape 4

Créer le fichier **hpasswd** et trois usagers :

```
# htpasswd -c /etc/httpd/conf/hpasswd tony
New password:
Re-type new password:
Adding password for user tony
```

```
# htpasswd  /etc/httpd/conf/hpasswd amelie
New password:
Re-type new password:
Adding password for user amelie
```

```
# htpasswd  /etc/httpd/conf/hpasswd anne
New password:
Re-type new password:
Adding password for user anne
```

Étape 5

Modifier le fichier **httpd.conf**

```
vi /etc/httpd/conf/httpd.conf
```

```
    <Location /achat>
        AuthType Basic
        AuthName "Données des achats"
        AuthUserFile /etc/httpd/conf/hpasswd
        AuthGroupFile /etc/httpd/conf/hgroup
        require group marketing
    </Location>
```

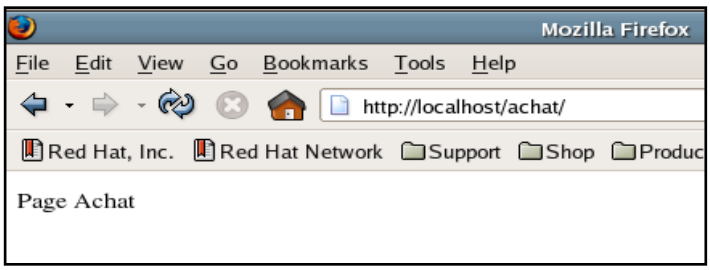
Étape 6

Redémarrer le service **apache** :

```
# service httpd restart
Stopping httpd:                                    [  OK  ]
Starting httpd:                                    [  OK  ]
```

Étape 7

Tester l'URL http//localhost/achat


Mozilla Firefox
File Edit View Go Bookmarks Tools Help
http://localhost/achat/
Red Hat, Inc. Red Hat Network Support Shop Produc
Page Achat