SERVICE APACHE :

SÉCURITÉ ET CONTRÔLE D’ACCÈS

Serveur Linux CentOS

Table des matières

1 Sécuriser un serveur Apache

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.

1.1 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

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

```
[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.
```

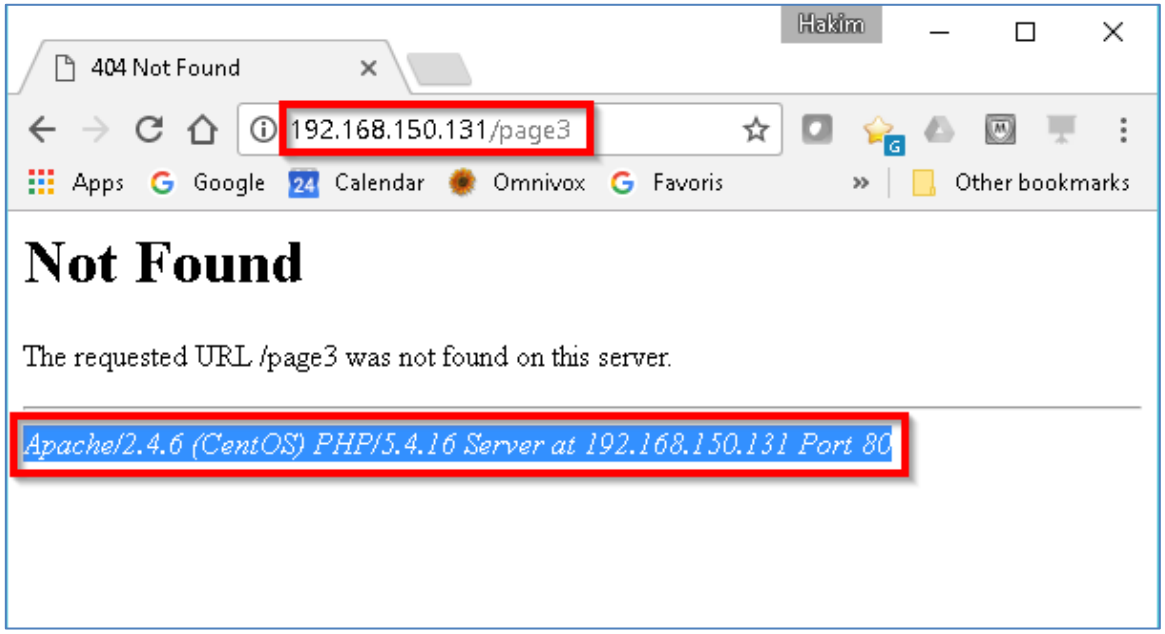
Ainsi, la bannière : `Server: Apache/2.4.6 (CentOS) PHP/5.4.16` se limite à la bannière : `Server: Apache`

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

1.2 Désactiver la signature

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

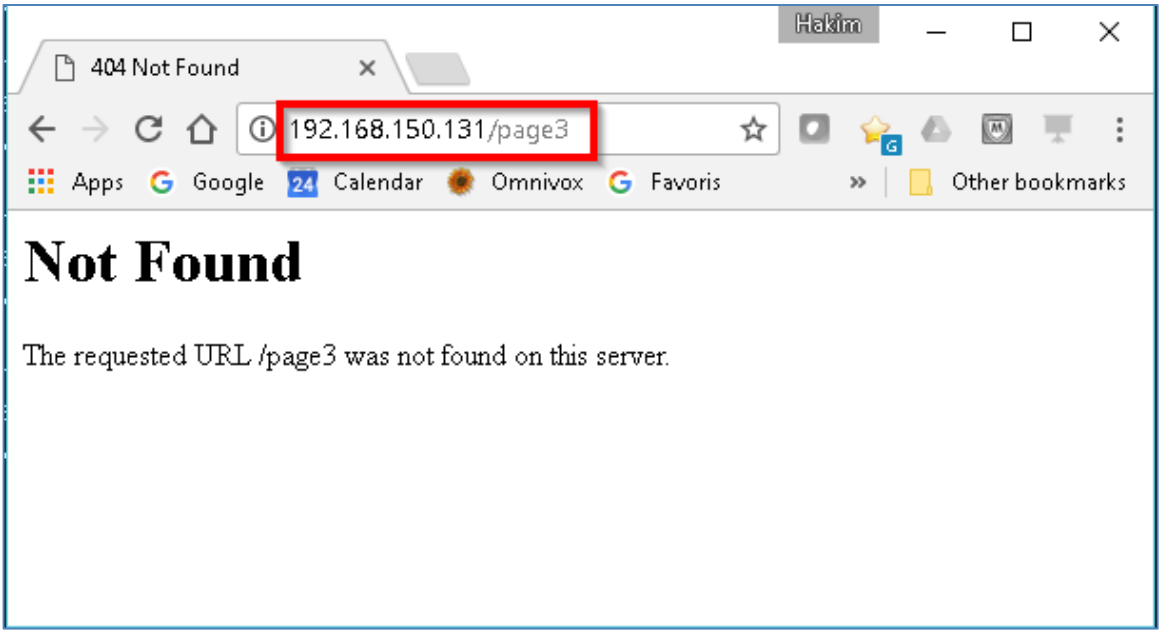
Si vous demandez une page inexistante, **apache** renvoie une page d'erreur **404** avec en bas de la page, le message *Apache/2.4.6 (CentOS) PHP/5.4.16 Server at 192.168.150.131 Port 80* qui révèle la version du service **apache**.



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

```
ServerSignature Off
```

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



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

1.3 Contrôle d’accès

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

1.3.1 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.

1.3.2 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.

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**

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 **172.16.0.0/24** sauf **172.16.0.25**

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