Service web sécurisé: HTTPS

introduction

- Puisqu'il est accessible depuis l'extérieur, le serveur Web est également une cible privilégiée pour les pirates. Le plus souvent, ces derniers cherchent à nuire à l'image de marque de leur cible en modifiant la page d'accueil du site.

Gestion des droits

- La gestion des se réalise avec le module mod_access, il existe 3 catégories:
- **Directory**: qui désigne un répertoire
- **Location**: qui désigne une arborescence
- **File**: qui désigne un fichier
- Exemple:

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

Les directive d'apache

- **Log-Level**: niveau d'alete des logs
- **Redirect**: redirige les requetes vers une autre adresse
- **DocumentRoot**: le répertoire de base
- **FollowSymLinks** ou **AllowOverride**: indique au serveur de remplacer les directives définis dans .htaccess par les directives précédentes.
- +|-**Indexes** : permet de voir (ou non) le contenu du répertoire en question

Protection par mot de passe

- Le module mod_auth protège l'accès à un répertoire par mot de passe

```
<Location "/reprtoire">
    AllowOverride None
    AuthName "accès personnel"
    AuthType Basic
    AuthUserFile "/etc/apache/.passwd"
    Require valid-user
</Location>
AccessfileName .htaccess
<Files  -  "^\.htaccess">
   Order deny,allow
   Deny from all
</Files>
```

Création du fichier de mots de passes

- htpasswd –c /etc/apache/.passwd ali
- htpasswd /etc/apache/.passwd mina

Protection de fichiers dans

- <Files info.php>

  AllowOverride None

  AuthName "accès personnel"

  AuthType Basic

  AuthUserFile "/etc/apache/.common-users"

  Require valid-user
- </Files>

Limitation contre le déni de service

- MaxClient: limite le nombre de connexions simultanées
- KeepAlive on # active limitation
- MaxKeepAliveRequests 70 # nombre de demande de connexions max
- MaxAliveTimeout 5 #temps max de réponse

Installation de HTTPD

- Les paquetages apache2 et mod_ssl sont nécessaires au fonctionnement en mode sécurisé utilisant le chiffrement et l'authentification forte.

Directives pour le module mod-ssl

SSLEngine on|off ; activer/désactiver module ssl

SSLCertificateFile certificat du serveur

SSLCertificateKeyFile bi-cles du serveur

SSLCACertificateFile certificat de l'AC pour vérifier les certificats des clients

SSLCARevocationFile certificat de révocation

SSLVerifyClient require|optional_no_ca|none; imposer le certificat x509 du client/ peut présenter/non

SSLRequire (expression) paramètres imposés par le serveur pour les certificat

SSLPassPhraseDialog builtin; boite de dialogue pour saisir phrase passe de la clé au démarage

SSLRequireSSL; oblige le client à utiliser SSL (HTTPS)

configuration d'Apache2

- Il est nécessaire dans un premier temps d'activer le module ssl :
- a2enmod ssl
- ==> le répertoire /etc/apache2/mods-enabled/ contient : ssl.conf ssl.load
- /etc/apache2/site-available/default doit : NameVirtualHost * devient NameVirtualHost *:80 <VirtualHost *> devient <VirtualHost *:80> pour chaque virtualhost <VirtualHost adr_IP> devient <VirtualHost adr_IP:80> pour chaque virtualhost
- /etc/apache2/site-available/default-ssl doit:

NameVirtualHost *:443

<VirtualHost *:443>

Exemple de /etc/apache2/site-available/default-ss

```
NameVirtualHost *:443
<VirtualHost *:443>
ServerName abc.domain1.org
DocumentRoot /home/web/abc
#Les directives pour les certificats
SSLEngine On
SSLCertificateFile /etc/apache2/ssl/servabc.crt
SSLCertificateKeyFile /etc/apache2/ssl/servabc.key
#Les directives qui suivent ne sont pas obligatoires mais
#aident au débugage
#On isole les erreurs relatives à http
#Attention : il faut créer le fichier "error_ssl.log"
ErrorLog /var/log/apache2/error_ssl.log
LogLevel warn
</VirtualHost>
```

Activation du site

- Il est nécessaire maintenant d'activer le site : **a2ensite default-ssl**
- Tester**:**

✓ Lynx 'https://abc.domain1.org:443'

✓ openssl s_client -connect abc.domain1.org:443

Amélioration des performances SSL

- **Forcer le contenu à être desservi par SSL**

<VirtualHost 192.168.1.4:80>
ServerName private.exemple.com
Redirect / https://private.exemple.com/
</Virtualhost>

- **Authentification à l'aide des certificats client**

SSLVerifyClient require
SSLCACertificateFile \
/usr/local/ssl/openssl/certs/ca.crt

Contrôle d'accès complexe avec *mod_ssl*

```
SSLRequire ( %{SSL_CIPHER} !~ m/^(EXP|NULL)-/ \
and %{SSL_CLIENT_S_DN_O} eq "Snake Oil, Ltd." \
and %{SSL_CLIENT_S_DN_OU} in {"Staff", "CA",
    "Dev"}\
and %{TIME_WDAY} >= 1 and %{TIME_WDAY} <= 5 \
and %{TIME_HOUR} >= 8 and %{TIME_HOUR} <= 20 ) \
or %{REMOTE_ADDR} =~ m/^192\.168\.1\.[0-9]+$/
```