Apache
Créer un serveur Web « sécurisé »

Roger Yerbanga
contact@yerbynet.com

Transfer 1.3 - Bobo-Dioulasso - Décembre 2012

Apache & mod_ssl

- Apache est un serveur HTTP
  - libre, populaire et très apprécié
  - mod_ssl
  - module permettant d'utiliser SSL dans Apache
- Ce module est inclus dans Apache 2
  - et notre cours utilisera Apache 2 !

Installer Apache (1)

- Le faire soi-même
  - http://httpd.apache.org/
  - permet d'être à niveau des versions
- Les systèmes d'exploitation
  - disponible prêt à l'emploi avec presque tous les systèmes
  - pas forcément dans les versions les plus récentes
  - il y a des avis de sécurité de temps en temps…

Installer Apache (2)

- OpenSSL doit être installé au préalable !
- Pour le faire soi-même à partir des sources

  tar zxvf apache2*tar.gz

  cd apache*

  ./configure –enable-mods-shared=’ssl all’

  make && make install
- Il ne reste plus qu’à configurer…

Installer Apache (3)

- Fichier de démarrage
  - /etc/init.d/apache (ou similaire)
- Programme apachectl (ou apache2ctl)
  - avec ou sans SSL : apachectl start
- Clé & certificat X.509
  - si vous utilisez les SSL il faut avoir ces données avant le démarrage d'Apache

==> Commande : *a2enmod ssl* pour activer le module ssl si nécessaire

Sites Web « virtuels »

- Mots anglais : « virtual hosting »
  - – utiliser une instance d'Apache pour héberger plusieurs sites Web
  - – voire plusieurs centaines ou milliers de sites
- Deux modes avec Apache
  - – par IP : une adresse IP (ou un numéro de port) par site
  - – par nom : plusieurs sites se partagent la même IP

Sites virtuels par nom

- Très économique, à préférer
  – quelques très très anciens clients Web pourraient ne pas s'en sortir : tant pis pour eux
  – facile à configurer et fiable
  – une IP, port 80 peut héberger des centaines ou milliers de sites Web
  – chaque site est séparé des autres
- Ne convient pas aux Web sécurisés (HTTPS)

Sites virtuels par IP

- Nécessite beaucoup de ressources
  – un couple IP:port par site
- Indispensable pour les Web SSL
  – parce que l'échange des certificats se fait dès le début de la connexion SSL (avec le HTTP)
  – le certificat est lié au nom du serveur, Apache ne peut donc pas par avance savoir quel certificat utiliser

Si une seule IP est disponible…

- Utiliser des numéros de ports séparés
  - port 80 : tous les serveurs standard
  - port 443 : le « privilégié » en https
  - port 8443 : un autre site https
  - …
- Risque pour l'accès aux sites
  - hélas de nombreuses politiques de sécurité d'entreprises interdisent l'accès aux sites Web ailleurs que sur les numéros de ports standard (80 et 443)

DNS & Sites Web

- En plus de la configuration Apache, il faut un enregistrement DNS adéquat pour renvoyer le trafic
- Proposition :

– utiliser pour chaque site un enregistrement de type A (www.mondomaine.bf IN A 1.2.3.4) plutôt qu'un CNAME

– utiliser l'IP du serveur

– il n'y a pas de numéro de port : c'est dans l'URL que cela se trouve

Questions

Configuration d'Apache (1)

- Un seul fichier de configuration (httpd.conf ou apache.conf)
- Peut être subdivisé en plusieurs fichiers sur différents répertoires.
- Une structure simple à comprendre avec des sections

  <Nom de la section>

  contenu de la section

  </Nom de la section>

Configuration d'Apache (2)

- Nombre de serveurs à lancer (StartServers, MinSpare-Servers, …)
  – à adapter en fonction de l'activité du site
  – petit site : prendre des valeurs très basses (StartSer-vers = 1 par exemple)
- Listen 80
  – indique sur quelle(s) IP et numéros de ports Apache doit accepter les connexions
  – il faut indiquer toutes les combinaisons qui seront utili-sées

```
Listen 80
Listen 443
```

Configuration d'Apache (3)

- User / Group
  – ne pas oublier de ne pas faire tourner Apache sous l'identité privilégiée !
- HostnameLookups Off
  – il est conseillé de conserver cette option : sinon délais supplémentaires liés au DNS significatifs
  – les logiciels de statistiques s'occuperont de cela plus tard

Configuration d'Apache (4)

- NameVirtualHost *:80

  – permet de définir les adresses et numéro de ports utilisés pour de l'hébergement virtuel par nom

  – toutes les autres adresses sont par définition « virtuelles par adresse »

  – n'a pas de sens pour HTTPS (c'est à dire *:443 par exemple)

- Il faut ensuite définir les VirtualHost associés

Configuration VirtualHost (1)

- Serveur virtuel basé sur le nom

```
<VirtualHost *:80>
ServerAdmin admin@mon-domaine.bf
DocumentRoot /var/www/www.mon-domaine.bf/htdocs
ServerName www.mon-domaine.bf
ServerAlias web.mon-domaine.bf
ErrorLog /var/log/apache2/www.mon-domaine.bf/error_log
CustomLog /var/log/apache2/www.mon-domaine.bf/access_log common
</VirtualHost>
```

- Autant d’entrée que de NameVirtualHost

Configuration VirtualHost (2)

- Choix du VirtualHost « nommé »
  - – la liste est balayée séquentiellement
  - – dès que le nom correspond (ServerName ou ServerAlias) on s'arrête
  - – si le nom n'est pas trouvé, c'est la 1ère entrée qui sera utilisée
- Attention au nombre de fichiers ouverts !
  - – 2 fichiers par site rien que pour les journaux

Configuration VirtualHost (3)

- Serveur virtuel basé sur l'adresse IP

```
<VirtualHost 1.2.3.4:80>
ServerAdmin admin@mon-domaine.bf
DocumentRoot /var/www/mon-domaine.bf/
ServerName www.mon-domaine.bf
ServerAlias web.mon-domaine.bf
ErrorLog /var/log/apache2/mon-domaine.bf/error_log
CustomLog /var/log/apache2mon-domaine.bf/access_log common
</VirtualHost>
```

- Exactement une entrée par adresse IP/Port

  – ne pas oublier de mettre le « Listen » qui va bien

Configuration SSL

- Un VirtualHost par IP/Port

```
<VirtualHost 1.2.3.4:443>
ServerAdmin admin@mon-domaine.bf
DocumentRoot /var/www/mon-domaine.bf/
ServerName www.mon-domaine.bf
ServerAlias web.mon-domaine.bf
ErrorLog /var/log/apache2/mon-domaine.bf/error_log
CustomLog /var/log/apache2/mon-domaine.bf/access_log common
SSLEngine On
SSLCipherSuite ALL:!ADH:!EXPORT56:RC4+RSA:+HIGH:+MEDIUM:
    +SSLv2:+EXP:+eNULL
SSLCertificateFile /local/apache/ssl.crt/www.mon-domaine.bf.pem
SSLCertificateKeyFile /local/apache/ssl.key/www.mon-domaine.bf.pem
</VirtualHost>
```

- Cf. fichier ssl.conf d'Apache pour le reste des options & exercices

Création de certificat avecSSL

- Création de certificat auto-signé
- # Génération de clé avec RSA
- openssl genrsa -out rsa-privkey.pem 2048
- # Génération de certificat
- openssl req -new -x509 -key rsa-privkey.-pem -out cacert.pem -days 1095
- # Génération de clé avec DSA
- openssl dsaparam -out dsaparam.pem 2048
- openssl gendsa -out dsa-privkey.pem 2048 dsaparam.pem

Création des fichiers et répertoires

- Les répertoires doivent exister
  - mkdir -p /var/www/mon-domaine.bf/
  - mkdir /var/log/apache2/mon-domaine.bf/
- Et avoir les bonnes permissions
  - www : celui qui met à jour le site, lisible pour Apache
  - log : root ?
- Ne pas oublier de faire « tourner » les fichiers journaux

Dernières vérifications

- apachectl configtest
  – permet de vérifier la bonne syntaxe du fichier de configuration
- apachectl graceful
  – recharge le fichier de configuration sans interrompre brutalement les connexions en cours
- apachectl restart
  – la même chose en plus violent
- apachectl start / stop
  – arrêt et redémarrage d'Apache

Créer son autorité de certification ?

- Possibilité 1 : non
  – les programmes classiques (par exemple Apache + mod_ssl) savent générer avec OpenSSL un certificat auto-signé
  – c’est facile mais pas homogène
- Possibilité 2 : oui
  – assure un minimum d’homogénéité dans votre entité
  – c’est rigolo et intéressant de manipuler et comprendre ces outils

Questions

Exercices avec Apache

- Création de trois virtual hosts
- Dans chaque virtual host, on doit avoir un fichier html qui affiche le nom de domaine et le numéro d'ordre du virtual host.
- Tester les accès et vérifier les logs
- Ajouter un HTTPS pour l'un des virtual host
- Essayer de rajouter un deuxième HTTPS

Alias & Redirections

- Modules :
  - **mod_alias** : pour les redirections (les cas simples)
  - **mod_rewrite** : réécriture (pour les choses sérieuses)
- mod_alias : permet l'utilisation de Alias, ScriptAlias, et Redirect
  - Alias & ScriptAlias :
  - mappage entre URL et répertoire sur le serveur.
  - Permet d'utiliser des répertoires hors de la racine d'Apache.
  - ScriptAlias : pour les scripts
  - **Redirect** : averti le client de refaire une nouvelle requête avec une nouvelle URL.

Alias & Redirections

- Exemples :
  - 1. Alias /image /ftp/pub/image
  - 2. Alias /icons/ /usr/local/apache/icons/
  - Pourquoi **/icons** ne sera jamais aliasisé ?
  - AliasMatch ^/image/(.*)$ /ftp/pub/image/$1
  - AliasMatch ^/image/(.*)\.jpg$ /files/jpg.images/$1.jpg
  - AliasMatch ^/image/(.*)\.gif$ /files/gif.images/$1.gif
  - AliasMatch : utilisation d'expressions régulières
  - ScriptAlias /cgi-bin/mailman /usr/lib/cgi-bin/mailman
- Utilisation de toutes ces directives entraîne forcément une déclaration de Directory, pour définir les accès au répertoire en question.

Alias & Redirections

- ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options +ExecCGI
    Order allow,deny
    Allow from all
  </Directory>
- Alias /image /ftp/pub/image
  <Directory /ftp/pub/image>
     Require all granted
  </Directory>

Alias & Redirections

- Redirect et RedirectMatch
- Différences entre Alias et AliasMatch va-lables
- Exemples :
  - # Rediriger vers un hôte différent
  - Redirect /service http://foo2.example.com/service
  - Redirect / http://www.yerbynet.com/
  - # Rediriger vers le même serveur
  - Redirect /ouaga /bobo
  - RedirectMatch ^/$ /backuppc/
  - RedirectMatch ^/$ https://ns.refer.sn/
  - RedirectMatch (.*)\.gif$ http://other.example.com$1.jpg

Réécriture

- DocumentRoot /var/www/example.com
  Alias /myapp /opt/myapp-1.2.3
  <Directory /opt/myapp-1.2.3>
  RewriteEngine On
  RewriteBase /myapp/
  RewriteRule ^index\.html$ welcome.html
  </Directory>
- <Location /squirrelmail>
  RewriteEngine on
  RewriteCond %{HTTPS} !^on$ [NC]
  RewriteRule . https://%{HTTP_HOST}%{REQUEST_URI} [L]
  </Location>

Autorisation et contrôle d'accès

- **Autorisation par hôte (IP, nom) ou réseau**

- Module : **mod_access_compat**
- # Par nom
  Allow from example.org
  Allow from .net example.edu
- # Par IP
  Deny from 10.1
  Deny from 10.172.20 192.168.2
- Allow from 10.1.0.0/255.255.0.0

- <Directory />
  Order deny,allow
  Deny from all
  Allow from 41.8.1.3 10.6.3.
  2002:d59a:4141:ba0::/64
  </Directory>
- Les directives fournies par mod_access_compat sont obsolètes
- Utiliser désormais (version 2.4+) **Require** de **mod_authz_host**
- Require ip 192.168.1.104 192.168.1.205

Autorisation et contrôle d'accès

- **Autorisation par utilisateur**

- Plusieurs méthodes possibles :
- htpassword
- Utilisateurs du système
- Base de données
- LDAP
- ...

Autorisation et contrôle d'accès

- **Avec htpassword :**
- <Directory "/var/www/html/vacance">
  AllowOverride AuthConfig
  Order allow,deny
  Allow from all
  AuthType Basic
  AuthName "Test de protection"
  AuthUserFile /etc/httpd/conf.d/.passwd.vacance
  Require valid-user
  </Directory>
- htpasswd -bc /etc/httpd/conf.d/.passwd.vacance roger monpasswd

Autorisation et contrôle d'accès

- **System users :**
- <Directory "/var/www/html/vacance">
  AllowOverride None
  Order allow,deny
  Allow from all
  AuthPAM_Enabled on
  AuthType Basic
  AuthName "Test de protection"
  <Limit GET POST>
  Require valid-user
  </Limit>
  </Directory>

Autorisation et contrôle d'accès

- **MySQL DB :**
- <Directory "/var/www/html/vacance">
  AllowOverride All
  AuthType Basic
  AuthName "Test de protection"
  AuthMYSQLEnable on
  AuthMySQLDB vacusersdb
  AuthMySQLUser vacuseradm
  AuthMySQLPassword vacpasswd
  AuthMySQLUserTable vacusers
  AuthMySQLNameField username
  AuthMySQLPasswordField password
  # AuthMySQLCryptedPasswords On
  Require valid-user
  </Directory>

Exemple de virtual host avec ssl

- <VirtualHost 10.1.2.3:443>
  ServerAdmin webmaster@localhost
  DocumentRoot /var/www
  <Directory /var/www/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride None
    Order allow,deny
    allow from all
  </Directory>
  ErrorLog ${APACHE_LOG_DIR}/ssl_error.log
  LogLevel warn
  CustomLog ${APACHE_LOG_DIR}/ssl_access.log combined
  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
  SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
  </VirtualHost>

Questions

Exercice

- Authentifier l'accès à l'un des sites précé-demment créé en utilisant la méthode « ht-password »

- Installer squirrelmail
- Le faire tourner uniquement en https quelque soit l'url entrée par l'utilisateur
- Restreindre l'accès à votre seule machine