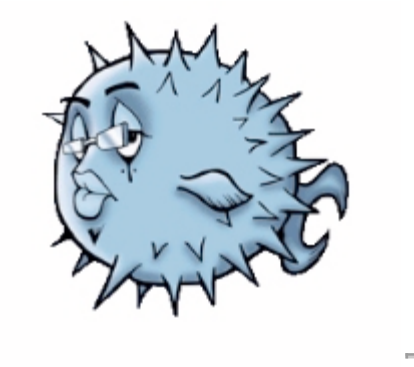

Apache : LE serveur Web

1 Serveur Web

User www ***group www*** Identité sous laquelle tourne Apache dès que le serveur s'est "accroché" (bind to) aux ports privilégiés (< 1024) standards 80 (http) et 443 (https) pour lesquels un privilège superutilisateur est nécessaire.

Depuis la 3.2 httpd, le démon Apache est chrooté (mis en prison) dans /var/www. Ceci augmente drastiquement la sécurité de la mchine en cas de compromission du service web, l'attaquant se retrouve avec l'identité www.www dans une prison dont la racine virtuelle est /var/www.

httpd.conf

Section 1:Global Environment

- ServerAdmin Mon_email

Section 2 : Main server configuration

- DirectoryIndex index.html **index.php** Si vous voulez éviter qu'en l'absence d'index.html dans un répertoire de script php, vous vous retrouviez avec la liste des fichiers présents..
- ServerSignature On **Off** Email
  Qu'est ce qui apparait en cas d'erreur 404 ? le nom et la version d'apache / Rien / L'email de l'administrateur

Gestion des hôtes virtuels basés sur le nom
Apache répond, évidemment , aux normes http 1.1; il est capable d'héberger et de répondre à plusieurs noms de machines, de domaines avec un nombre réduit d'adresses IP :
Vous n'avez qu'une seule adresse IP fixe publique (Ce n'est deja pas mal!) et vous voulez heberger deux (ou plus) serveurs répondant à des noms radicalement différents. Pour cela vous devez utiliser la notion d'hôte virtuel. Attention les vieux clients http 1.0 ne sauront pas résoudre ce nom.

Dans la section 3: Virtual Hosts, ajoutez les lignes suivantes :
#Evidemment vous rangerez cela comme vous voulez chez vous..
#<VirtualHost _default_:*>
#</VirtualHost>

######Config de mon premier serveur virtuel

<VirtualHost Mon_IP_Publique > Définition du bloc d'instructions relatives au premier hote virtuel

ServerAdmin email-1 Adresse admin de cet hôte virtuel
ServerName www.mon_domaine-1 Son nom DNS
ServerAlias *.mon_domaine-1 Intercepte toute connexion vers
n"importe_quoi.openbsd-edu.net
DocumentRoot /var/www/mon_domaine-1/ Les documents publiés par cet hôte
ScriptAlias /cgi-bin/ /var/www/mon_domaine-1/cgi-bin/ Les scripts CGI qui li sont
particuliers.
TransferLog /var/www/mon_domaine-1/logs/access_log Ses logs d'accès
ErrorLog /var/www/mon_domaine-1/logs/error_log Ses logs d'erreur
</VirtualHost>

######Config de mon second serveur virtuel

<VirtualHost Mon_IP_Publique >
ServerAdmin email-2
ServerName www.mon_domaine-2
ServerAlias *.mon_domaine-2
DocumentRoot /var/www/mon_domaine-2/
ScriptAlias /cgi-bin/ /var/www/mon_domaine-2/cgi-bin/
TransferLog /var/www/mon_domaine-2/logs/access_log
ErrorLog /var/www/mon_domaine-2/logs/error_log
</VirtualHost>

######Config de mon troisième serveur virtuel ***en HTTPS***

<VirtualHost Mon_IP_Publique:443 >
ServerAdmin email-3
ServerName www.mon_domaine-3
ServerAlias *.mon_domaine-3
DocumentRoot /var/www/mon_domaine-3/
ScriptAlias /cgi-bin/ /var/www/mon_domaine-3/cgi-bin/
TransferLog /var/www/mon_domaine-3/logs/access_log
ErrorLog /var/www/mon_domaine-3/logs/error_log

Les hotes virtuels en ssl doivent inclure dans leur définition la partie de fichier (plus loin) gérant ces définitions.

</VirtualHost>

Attention!!! : On ne peut pas héberger plus de un hote virtuel https par adresse IP.

Vérification de la configuration :
apachectl configtest

Pour relancer apache :
apachectl restart

Pour relancer apache le plus doucement possible :
apachectl graceful

2 Un serveur https en 3 lignes et 10 minutes !

Apache sait délivrer des flux cryptésTLS simultanément à des flux non cryptés. Les uns sur le port 443, les autres sur le classique port 80.
NB : TLS (Transport Layer Security, Sécurité de la couche transport) est la version de l'IETF du protocole SSL (Secure Socket Layer, Couche XXX sécurisée) inventé par Netscape.

Tout se fait dans /etc/ssl/private.
1. On crée la clé RSA 1024 bits du serveur :

openssl genrsa -out /etc/ssl/private/server.key 1024

Si on préfère un clé chiffrée par une passphrase(mais alors il faut être devant le serveur au démarrage..), :

openssl genrsa -des3 -out /etc/ssl/private/server.key 1024

2. Il faut maintenant créer la requête signée de certification utilisée pour la demande de certification auprès d'une autorité de certification. Préparez votre portefeuille...

openssl req -new -key /etc/ssl/private/server.key -out /etc/ssl/private/server.csr

Il vous sera demandé successivement :
Country Name (2 letter code) []:
State or Province Name (full name) []:
Locality Name (eg, city) []:
Organization Name (eg, company) []:
Organizational Unit Name (eg, section) []:
Common Name (eg, fully qualified host name) []:*Le nom DNS du serveur* ***Indispensable***
Email Address []:*L'adresse du hostmaster de votre serveur* ***Indispensable***

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:*Entrée*
An optional company name []:*Entrée*

3. Puis vous envoyez le fichier server.csr à Thawte ou autre autorité de certification.Si vous n'avez pas les moyens de sortir la centaine de dollars annuels, vous pouvez utiliser des certificats auto-signés. Le seul inconvénient est que vous n'êtes pas une autorité de certification, les navigateurs émettront un message d'alerte (désactivable pour partie) lors de la consultation de votre site.
Pour auto-signer :

openssl x509 -req -days 365 -in /etc/ssl/private/server.csr -signkey /etc/ssl/private/server.key -out /etc/ssl/server.crt

4. Puis modifiez */etc/rc.conf* en *httpd="-DSSL"*
Si vous voulez profitez du https sans avoir à redémarrer, : ***/usr/sbin/apachectl startssl***. (un simple apachectl restart n'est pas suffisant!!).
Votre machine est maintenant accessible par https://votre_machine sur le port 443.

On répète : Attention!!! : On ne peut pas héberger plus de un hote virtuel https par adresse IP.

Remarques sur la sécurité :

1. Attention aux dénis de service (DoS) !
Le dialogue entre le serveur et le client étant chiffré, Toutes les données (pas les en-têtes) doivent être chiffrées/déchiffrées par chacune des parties. Ceci induit une forte surcharge CPU (D'un facteur 10, voire plus entre un apache https et un apache http). Or, si, normalement, un client ne gère qu'un seul flux https simultané, il n'en est pas de même du serveur. Il peut s'effondrer avec seulement une vingtaine de requêtes simultanées. (Attention, il peut s'agir de simples paquets identiques envoyés de manière répétitive; le serveur essaiera systématiquement de la déchiffrer.)
Une solution pourrait être d'investir dans un de ces coûteux accélérateurs cryptographiques qu'OpenBSD sait gérer nativement, mais ce n'est qu'une protection d'échelle, il faudrait peut-être aller jusqu'à 200 connexions SSL simultanées pour écrouler votre serveur.
Le document d'origine.

2. Attention au passage que vous faites dans votre beau coupe-feu !
Le flux étant chiffré, il peut contenir n'importe quoi, en particulier des troyens, des tentatives de hack, des scans de port.....bref que des cochonneries que votre beau FireWall, ou votre bel IDS ne verront pas passer. Ceci, à moins de déchiffrer le flux avant de le faire arriver à la zone protégée.

En conclusion, n'utilisez pas https parce que c'est à la mode, mais parce qu'il répond à un réel besoin.
3

3. Webalizer

Webalizer est un analyseur de logs, très orienté logs de serveur Web, Apache au hasard...
Ca s'installe tout seul *(**cd /usr/ports/www/webalizer && make && make install &&& make clean**)*.
et ça se configure aussi facilement.(***cp /usr/local/share/examples/webalizer/sample.conf /etc/webalizer.conf***).

Pour lancer l'activation des requêtes inverses, afin de connaitre vos clients : ***webazolver -D DNSCache***. Ca peut être relativement long à 1 requete inverse par secone en moyenne.
Il est déconseillé de faire tourner webalizer sur des fichiers de logs en cours. Usez de la crontab pour faire tourner les logs et appliquer webalizer dessus.
10 */2 * * * ***/usr/local/bin/webalizer -D DNSCache -M 0 -g 1 -R 1*** Lance l'analyse toutes les 2 heures
10 2 */7 * * ***/usr/local/bin/webazolver -D DNSCache*** Recrée le fichier DNS inverse une fois par semaine

Le résultat
