Sécuriser un serveur Apache

SOMMAIRE

- Principaux risques
- Attaques DoS
- Paramétrage de PHP
- Mise en œuvre du protocole Https sur un serveur Apache
- Protéger avec Fail2Ban
- Surveiller et journaliser les accès
- Autorisations et restrictions: mise en œuvre de htaccess
- Mettre Apache en "jail"

Serveur web

- Un serveur web est un serveur informatique utilisé pour publier des sites web au public ou interne à une Entreprise. Seulement, ces serveurs web là s'exposent à des risques parfois ignorés qui sont importants et dangereux également.

Principaux risques

- Les serveurs web peuvent toujours faire l'objet d'attaque que ce soit par des virus ou par des pirates experts. Les conséquences sont constitués généralement par des fuites d'informations personnelles et par l'exploitation des données personnels de l'utilisateur personne physique ou morale. Une mauvaise configuration de sécurité peut produire une faille lorsque les serveurs d'application, serveurs web, serveur de base de données, et la plate-forme n'ont pas de configuration sécurisée correctement établie et déployée. Bref, le particulier ou l'Entreprise doit toujours rester vigilant pour éviter d'être victime d'une quelconque atteinte.

Attaque DoS

- Une attaque par déni de service est une attaque informatique ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser.

Attaque DoS

Se protéger contre les attaques DoS

On peut réduire le TimeOut d'apache à 60 secondes au lieu de 300 secondes (à adapter suivant si le site peut avoir des requêtes longues) :

Code TEXT :

```
Timeout 60
```

Mais aussi réduire le nombre de processus apache à lancer par le serveur :

Code TEXT :

```
ServerLimit 64
```

Paramétrage de PHP

- PHP a été conçu spécifiquement pour la programmation web en tenant compte d'impératif de sécurité. Il est grandement paramétrable. Il est conseillé de définir les options suivantes dans son fichier de configuration /etc/php.ini. safe_mode surveille les fichiers accédés et interdit l'usage de commande à risque, expose_php contrôle l'affichage de sa banner, max_execution_time et memory_limit limite les risques de DoS, magic_quotes_gpc ajoute des quotes pour les données reçues des GET/POST et des cookies. Enfin, souvent oublié en production, il faut éviter l'affichage des lignes fautives lorsque les scripts PHP plantent.

Paramétrage de PHP

- safe_mode = On
- expose_php = Off
- max_execution_time = 30 ; Maximum execution time of each script, in seconds
- memory_limit = 8M
- magic_quotes_gpc = On
- display_errors = Off
- [SQL]
- sql.safe_mode= On

Mise en œuvre du protocole Https sur un serveur Apache

- L'accès à un serveur WEB utilise le protocole http via le réseau internet. Il est possible qu'un pirate intercepte vos requêtes et les réponses faites par le serveur car les informations qui circulent sur le réseau ne sont pas cryptées.
- Le protocole HTTPS (http over SSL) permet de remédier à cela :

- Les échanges sont cryptés

- L'authentification SSL du serveur est mise en œuvre

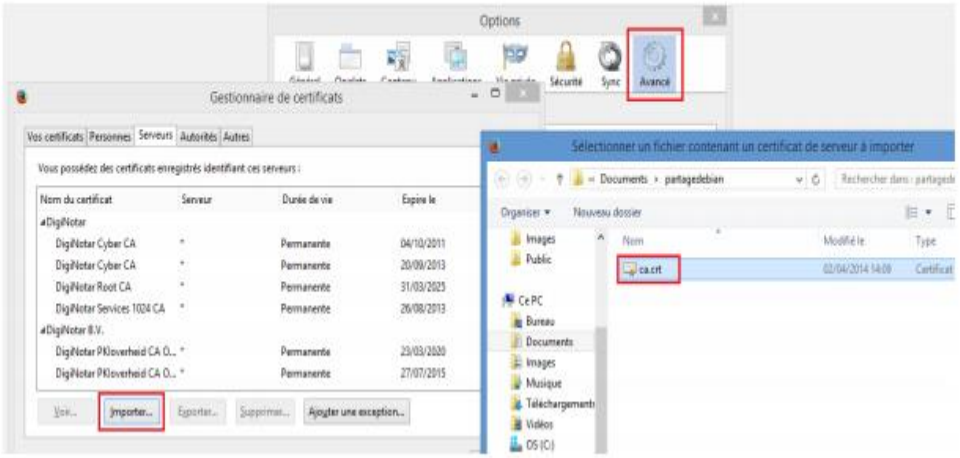
- Mettre en place le protocole SSL sur le serveur web

Mise en œuvre du protocole Https sur un serveur Apache

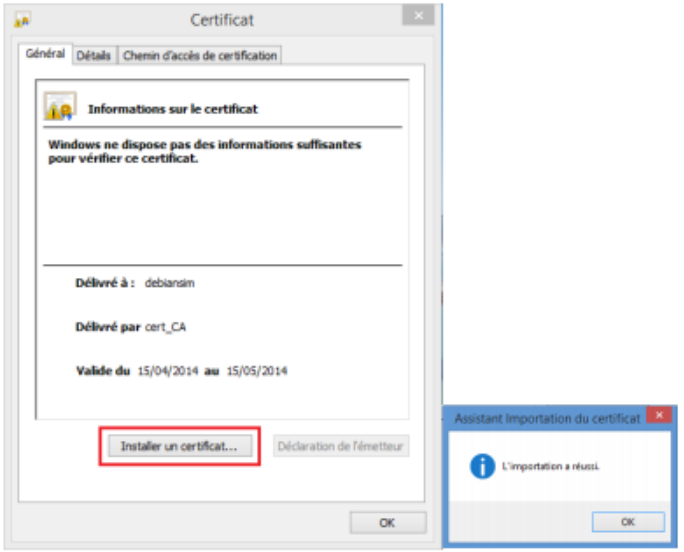
Il faut installer le fichier wwwssl.crt.

Sur mozilla :

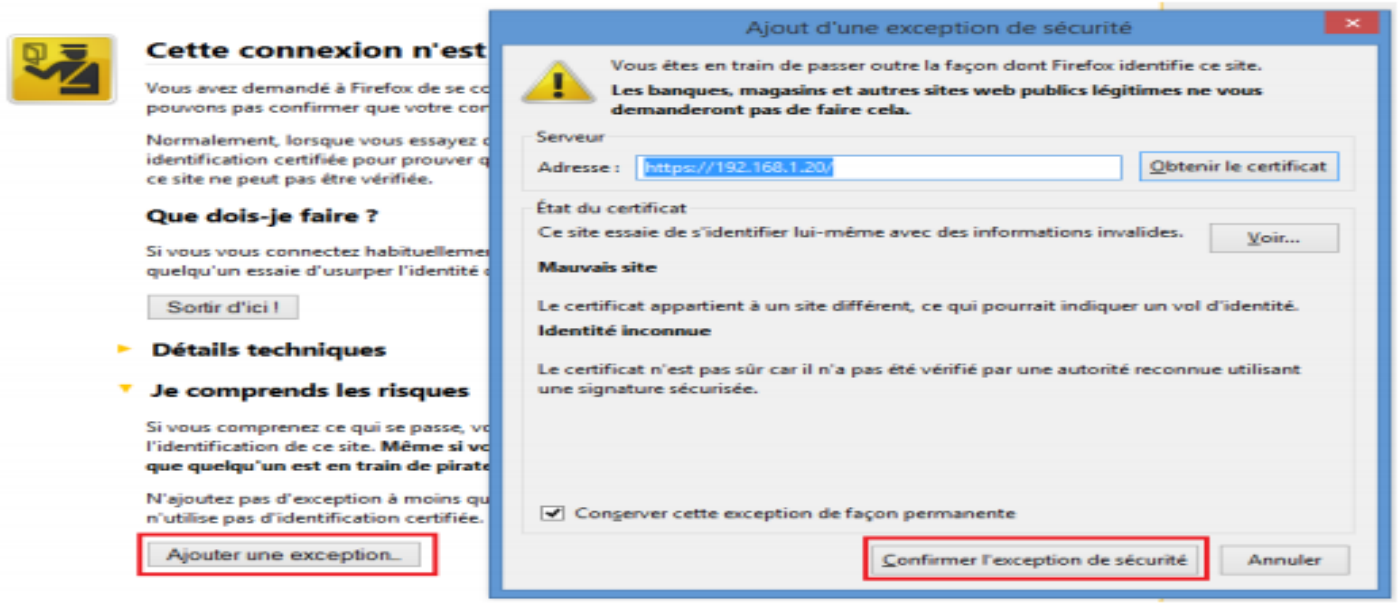
Aller dans Paramètres>Avancés>Afficher les certificats

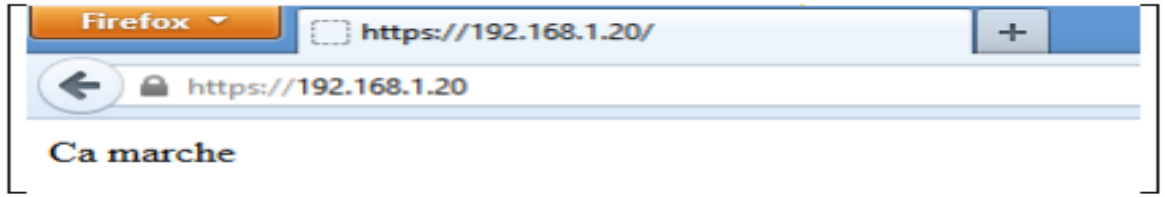


Il faut aussi installer ce certificat sur le poste client



Mise en œuvre du protocole Https sur un serveur Apache





Protéger avec Fail2Ban

Installer Fail2Ban

Une fois que votre serveur Apache est en cours d'exécution et que l'authentification par mot de passe est activée, vous pouvez installer «+ fail2ban +» (nous incluons ici une autre récupération de référentiel au cas où Apache aurait déjà été configuré aux étapes précédentes):

```
sudo apt-get update
sudo apt-get install fail2ban
```

Cela installera le logiciel. Par défaut, `+ fail2ban +` est configuré pour interdire les tentatives de connexion SSH ayant échoué. Nous devons activer certaines règles qui le configureront pour vérifier dans nos journaux Apache les modèles indiquant une activité malveillante.

Protéger avec Fail2Ban

Réglage des paramètres généraux dans Fail2Ban

Pour commencer, nous devons ajuster le fichier de configuration utilisé par `+ fail2ban +` pour déterminer les journaux d'application à surveiller et les actions à entreprendre lorsque des entrées incriminées sont trouvées. Le fichier `+ / etc / fail2ban / jail.conf +` fourni est la principale ressource fournie à cet effet.

Pour apporter des modifications, nous devons copier ce fichier dans `+ / etc / fail2ban / jail.local`. Cela évitera que nos modifications ne soient écrasées si une mise à jour du package fournit un nouveau fichier par défaut:

```
sudo cp /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

Ouvrez le fichier récemment copié afin que nous puissions configurer notre surveillance de journal Apache:

```
sudo nano /etc/fail2ban/jail.local
```

Surveiller et journaliser les accès

- Le journal des accès au serveur enregistre toutes les requêtes que traite ce dernier. La localisation et le contenu du journal des accès sont définis par la directive LogFormat permet de simplifier la sélection du contenu du journal. Cette section décrit comment configurer le serveur pour l'enregistrement des informations dans le journal des accès.

Surveiller et journaliser les accès

Format habituel du journal

Voici une configuration typique pour le journal des accès :

```
LogFormat "%h %l %u %t \"%r\" %>s %b" common
CustomLog logs/access_log common
```

mise en œuvre de htaccess

Créer un fichier .htaccess dans le répertoire que vous voulez protéger et mettez y ceci :

```
AuthUserFile /var/www/topsecret/.htpasswd
AuthGroupFile /dev/null
AuthName "Accès interdit aux citoyens de seconde zone..."
AuthType Basic
<Limit GET POST>
require valid-user
</Limit>
```

Mettre Apache en "jail"

5 - Création de la jail

Nous allons maintenant choisir dans sysinstall d'installer les sources de FreeBSD qui iront se mettre dans /usr/src. Une fois cela fait nous allons créer une jail. Commençons par expliquer ce qu'est une jail.
Une jail est une technique pour enfermer des processus dans un sous système de notre serveur. Le noyau reste le même mais la jail possède ses propres applicatifs (que l'on réduira au maximum) ainsi que des droits très limités en exécution. Les processus de la jail sont différencier de ceux de notre système. Ainsi un attaquant qui prend la main sur la jail sera restreint dans cette partie et le système principal continuera de fonctionner.

Revenons donc sur l'installation :
Nous allons créer un répertoire /jails a la racine et un repertoire www/ à l'intérieur de celui ci qui identifiera notre jail pour le serveur web. On va alors compiler tout le système de base de notre FreeBSD a l'intérieur de la jail par la commande :

```
cd /usr/src && make world DESTDIR=/jails/www
```

Cette commande prend un certain moment, selon la puissance de la machine. Une fois cela terminé on installe les fichiers de configuration de base via la commande

```
cd etc/ && make distribution DESTDIR=/jails/www
```

Mettre Apache en "jail"

Puis :

```
mount_devfs devfs /jails/www/dev
```

afin de monter les périphériques a l'intérieur de la jail

Notre jail est donc maintenant installer. Il va falloir maintenant la configurer.

On va activer alors la jail dans le /etc/rc.conf afin de pouvoir la lancer :

```
jail_enable="YES"   # Utiliser NO pour désactiver le lancement des environnements jail
jail_list="www"     # Liste des noms des environnements jail séparés par une espace
jail_www_rootdir="/jails/www"     # le répertoire racine de l'environnement jail
jail_www_hostname="gargamel"  # le nom de machine de l'environnement jail
jail_www_ip="172.16.3.100"          # son adresse IP
jail_www_devfs_enable="YES"         # monter devfs dans l'environnement jail
jail_www_devfs_ruleset= "devfsrules_jail"# les règles devfs à appliquer à l'environnement jail
```

On va également attribuer une IP fixe a notre serveur ainsi qu'un alias sur l'interface ayant l'adresse de la jail. On ajoute a /etc/rc.conf :

```
ifconfig_rl0="inet 172.16.3.103 netmask 255.255.255.0"
ifconfig_rl0_alias0="inet 172.16.3.100 netmask 255.255.255.0"
```

Mettre Apache en "jail"

Il ne reste plus qu'à lancer la jail via la commande :

```
/etc/rc.d/jail start
```

Sources

- http://www.access-management.info/a-quels-risques-les-serveurs-web-sont-ils-exposes/
- https://www.linuxtricks.fr/wiki/apache-installation-configuration-securisation
- https://www.cgsecurity.org/Articles/apache.html
- https://www.system-linux.eu/index.php?post/2009/04/23/Mettre-en-place-un-htaccess-avec-htpasswd
- http://blog.madpowah.org/articles/jail/index.html