Web et sécurité

Table ronde Ankeo lors de la réunion XXX - 22/06/2004

Web et sécurité....1
La société Ankeo....1
Principes généraux....1
Quelques pistes avec IIS....2
Quelques piste avec Apache....2
Exemple: la limitation des accès....2
Cas particulier de l'architecture LAMP....3
Liens utiles....3

La société Ankeo

Existant depuis 2001, nous sommes spécialisés sur tout ce qui touche à la sécurité informatique: de l'analyse, l'audit ou les tests d'intrusions à la formation en passant par le conseil et l'ingénierie (mise en place de firewalls, antivirus, etc.).

Web: http://www.ankeo.com

Adresse : 10 rue du Rhône - 68100 Mulhouse

Tél: 03 89 66 03 85

Jean-Marc Boursot - jmb@ankeo.com

Principes généraux

Tout ce qui n'est pas explicitement utile est totalement inutile.

La sécurité n'est pas un produit mais un processus, donc un élément à constamment suivre, analyser et remettre en cause.

La démarche de sécurisation ira du général au particulier:

- sécurisation du réseau et de l'architecture
  - ✔ firewall (limitation aux ports 80/tcp et/ou 443/tcp) ou reverse proxy
  - ✔ DMZ (zone démilitarisée)
  - ✔ cloisonnement
- sécurisation du système
  - ✔ mises à jour de sécurité
  - ✔ limitation des services installés ou qui tournent
  - ✔ droits utilisateurs/fichiers/répertoires
- sécurisation du service

- ✔ paramètres
- ✔ ajout d'une authentification
- ✔ travail en ssl (https)

- sécurisation de l'application
  - ✔ php, asp, cgi, etc.
  - ✔ base de données

Quelques pistes avec IIS

- Réseau et système
  - ✔ Serveur autonome et non dans un domaine.
  - ✔ Faire attention aux services annexes (SMTP, FTP et Gopher).
- Propres au service
  - ✔ Supprimer les sites web par défaut, les pages d'exemple, documentations et répertoires virtuels (Internet Services Manager) et les répertoires physiques correspondant (Explorateur).
  - ✔ Supprimer les mapping inutiles sur le serveur web.
  - ✔ Limiter les options et éléments de débugging.

Quelques piste avec Apache

- Les directives *ServerTokens Prod* et *ServerSignature off* pour limiter les infos recevables à distance concernant les modules et versions.
- Limitation des modules chargés et activés.
- Supprimer les alias et extentions inutiles.
- Attention aux options et droits des répertoires.

Exemple: la limitation des accès

On peut restreindre l'accès aux pages d'un répertoire soit par l'adresse d'origine du poste client, soit par une demande d'identification. Cela se fait de la manière suivante:

```
<Directory /httpd/html/repertoire_protege>
  # protection du répertoire par mot de passe
  AuthType basic
  AuthName "zone protégée"
  # fichier des mots de passe
  AuthUserFile /etc/apache/.users
  # règle d'authentification: il suffit d'avoir un utilisateur valide
  Require valid-user

  # limitation d'accès à notre plage d'adresse uniquement
```

```
  Order allow,deny
  Allow from 10.134
</Directory>
```

Ces directives peuvent également être placées dans un fichier `.htaccess` situé dans le répertoire à protéger. Dans ce cas, vérifier bien la présence d'une directive `AllowOverride AuthConfig Limit` ou `AllowOverride All` pour le répertoire en question (*AllowOverride* permet de "supplanter" certains paramètres du fichier de configuration pour un répertoire par ceux du fichier `.htaccess` de ce répertoire).

Le fichier `/etc/httpd/conf/.users` contenant les utilisateurs et leurs mots de passe est généré et modifié par la commande `htpasswd`.

Il faudra veiller à ce que les modules nécessaires soient dans la configuration, à savoir *mod_access* et *mod_auth*.

Cas particulier de l'architecture LAMP

Outre Apache, deux services essentiels - la base de données MySQL et le système de script PHP - tournent et il convient également de s'intéresser de près à leur sécurité. Ce point est le plus utilisé dans les attaques d'aujourd'hui, car plutôt que l'infrastructure ou le serveur, qui sont de mieux en mieux protégés, les attaquants préfèrent les applications très souvent vulnérables (la liste des applications PHP qui posent problème est énorme).

Attention car les éléments donnés ci-après ne sont que des pistes et ne peuvent suffire à sécuriser l'architecture LAMP.

- PHP (via le fichier php.ini)
  - ✔ travailler en mode protégé (safe_mode devrait être sur on)
  - ✔ faire attention aux variables globales (register_globals devrait être sur off)
  - ✔ limiter les fonctions autorisées (disable_functions)
  - ✔ limiter les messages d'erreur (display_errors)
  - ✔ limiter la réception de fichiers sur le serveur (file_uploads)
- MySQL
  - ✔ ajouter un mot de passe à l'utilisateur administrateur
  - ✔ ne jamais utiliser l'administrateur pour accéder à une base
  - ✔ créer une base par application, un utilisateur par base et gérer les droits

Liens utiles

Suite au différents sujets abordés lors de la table ronde, voici quelques liens qui permettront de poursuivre la réflexion.

http://www.php.net/manual/fr/print/security.index.php - sécurité de PHP (documentation officielle)

http://dev.mysql.com/doc/mysql/fr/Privilege_system.html - sécurité de MySQL (documentation officielle)

http://httpd.apache.org/docs/misc/security_tips.html - conseils de sécurité Apache (documentation officielle en anglais)

http://www.framasoft.net/article2138.html - présentation de la messagerie de Mozilla et plus particulièrement de son système antispam

http://gnuwin.epfl.ch/fr/index.html et **http://www.framasoft.net/** - logiciels libres pour Windows