TABLE DES MATIÈRES

INTRODUCTION

Comment éviter que votre site web ne soit utilisé par un pirate comme plate-forme de phishing (hameçonnage) ou d'envoi massif de spams (pourriels) ? Comment éviter le "defacing", c'est-à-dire l'effacement de votre site web et son remplacement par un autre, ou une page avec un slogan anti-occidental ? Comment éviter certains trous de sécurité ?

A priori, les serveurs mutualisés de votre hébergeur devraient être relativement sécurisés et disposer d'outils permettant de bloquer certains comportements suspects. Ce sont des professionnels qui ont la maîtrise de leurs outils. Ils mettent à votre disposition un espace que vous devez contrôler, ils ne le font pas pour vous. En effet, ils ne s'occupent que de leur partie (la gestion du matériel et de leurs serveurs), et vous êtes responsable des données que vous y mettez (site web, e-mail, fichiers…). Par conséquent, l'hébergeur suspend votre site web après une attaque, vous laissant le soin de corriger le problème tout seul. Tant que vous n'intervenez pas, cette action a posteriori bloque votre compte et votre site web reste inaccessible. Il est donc préférable de prévenir le piratage pour l'empêcher de vous nuire.

Voici donc quelques conseils concrets, pratiques et très efficaces. C'est l'accumulation de ces trucs et astuces qui sécurisera votre site, car il n'y a pas de solution unique; les pirates utilisent plusieurs moyens différents pour forcer un compte.

Qui sont les pirates ?

Les premiers sont des "skiddy", des jeunes ("kid" en anglais, "kiddy" pour petit jeune) qui utilisent des scripts prêts à l'emploi (le "s" de skiddy) qu'on trouve facilement sur le web pour exploiter les failles d'un CMS, blog, e-commerce, etc. Ils ne font qu'utiliser ces scripts comme on utilise un logiciel. Ce ne sont pas des "petits génies", ils ne programment pas et n'inventent rien. Ils se lancent des défis à celui qui effacera ou violera le plus de sites web. Les autres, bien plus dangereux, sont des pirates au service d'une mafia afin de prendre le contrôle de votre site web via une faille de votre CMS, blog ou e-commerce pour le convertir en plate-forme d'envoi de spams ou de phishing, ou en automate pour violer d'autres ordinateurs. Ceux-là créent leurs propres scripts qu'ils ne partagent pas avec une communauté. Ils font cela pour l'argent; les mafieux les payent en fonction du nombre de sites web piratés, d'identifiants et mot de passe récoltés, de moyens de paiements interceptés, d'ordinateurs personnels dont ils ont pris le contrôle à l'insu du malheureux propriétaire (c'est-à-dire votre PC et chez vous en profitant de votre connexion à internet en programmant un *malware* par exemple).

Pourquoi attaquent-ils votre site ?

Ni le skiddy, ni le mafieux ne vous visent personnellement. Les uns le font pour le jeu, les autres pour l'argent. Il est peu probable qu'on vous vise personnellement. Certains skiddies effacent des sites et se cachent derrière des pseudo slogans politiques et anti-occidentaux, histoire de vous faire peur, de se donner de l'importance et se prendre au sérieux. Il n'en est rien.

Comment savent-ils que mon site a une faille de sécurité ?

Réponse: Google ou tout autre moteur de recherche ! Il cherche un fichier précis comme login.php, confip.php ou autres, et, combiné avec quelques mots-clés, ils savent quel CMS, blog ou e-commerce vous utilisez. Ils essaieront alors de lancer un script pour tester si l'attaque fonctionne. Il ne font même pas ça manuellement, car ils ont des logiciels qui le font automatiquement !!! Leurs logiciels testent chaque URL listée par Google à la recherche de la faille. C'est aussi simple que ça. Ils vous trouvent par hasard.
Donc, nous allons essayer de nous prémunir contre ces attaques automatiques. Ces conseils ne concernent que les sites web utilisant un CMS, blog ou e-commerce, etc. programmés par des informaticiens ou par vous.

CONSEIL NUMÉRO 1: votre CMS, blog ou e-commerce doit être à jour. Vous suivrez les mises à jour de sécurité et les installerez sans attendre.

CONSEIL NUMÉRO 2: LES RÈGLES LES PLUS IMPORTANTES

Comme ce tutoriel est long, voici les règles qu'il faut appliquer en priorité. Vous pourrez inclure les autres plus tard.

1. Attribuer par FTP aux fichiers les droits *chmod 404* et aux dossiers les droits *chmod 505*. Voir l'article ci-dessous

2. Règles de filtrage par *htaccess*. Permet d'arrêter de nombreuses attaques avant de toucher votre site web. Voir l'article ici.

3. Règles de sauvegarde et de restauration de votre site web. D'abord, vérifiez quels fichiers le pirate a ajouté ou modifié en installant ce script. Ensuite, êtes-vous capable d'effacer complètement votre site web pour supprimer toutes les traces du pirate et de tout réactiver en 30 minutes? Voici comment. Lire l'article ici.

4. Si vous programmez en PHP, des règles simples de filtrages sont ici. Même chose pour l'injection SQL. Ne soyez pas irresponsable, pensez à filtrer tout ce qui entre dans votre script.

LES DROITS D'ÉCRITURE, DE LECTURE ET D'EXÉCUTION.

-= INDISPENSABLE =-

Plus d'infos ici: Description du CHMOD et de la signification des numéros.
Attention, ces règles peuvent fonctionner pour certains hébergeurs et pas chez d'autres. Faites un petit essai, cela en vaut vraiment la peine.
On a l'habitude de dire qu'on doit attribuer par FTP les droits 644 à un fichier et 755 à un dossier.
En fait, certains hébergeurs (mais pas tous) ne semblent pas utiliser de *groupe*. Donc, on pourrait très bien utiliser les droits 604 pour un fichier et 705 pour un dossier. Si un pirate pénètre le système avec un droit de *groupe*, il n'aura accès à rien, ni en lecture ni en écriture.
On peut aller plus loin. Protégeons les parties sensibles de votre CMS, blog ou e-commerce, comme le fichier config.php et .htaccess en lui donnant les droits 404 (ou 444). Personne ne pourra le modifier, même pas vous (c'est faux dans l'absolu si votre site a un gros trou de sécurité, mais imparable contre une attaque automatique). Vous ne pourrez le faire que par FTP quand vous aurez vraiment besoin de le modifier.

Voilà comment je protège mon site:

- - Tous les fichiers ont les droits 404 (ou 444).
- - Tous les dossiers ont les droits 505 (ou 555).
- - Si un fichier ou un dossier nécessitent des droits d'écriture par le serveur mettez 604 pour le fichier et 705 pour le dossier. Inutile de faire le fameux 777 (tous les droits à tout le monde) qui est un danger public, une provocation au piratage, car vous annoncez que votre maison est grande ouverte, sans porte ni fenêtre, tout le monde peut se servir.
- - Les fichiers *config* et *htaccess* ont des droits 404 (ou 444)
- - Le dossier "*www*" ou "*public_html*" **doit** être en *chmod 705* ou 755 selon votre hébergeur, ne le changez jamais.

Avantage: personne ne peut modifier vos fichiers.

Inconvénient: il faut changer les droits en écriture (644 et 755) si vous faites une mise à jour de votre CMS, blog ou e-commerce et remettre les bons droits 404 (ou 444) et 505 (ou 555) après. Cela prend 10 min., mais ça en vaut la peine. Si votre hébergeur ne vous permet pas de faire cela, déménagez chez un autre.

Pourquoi est-ce si important ?

Le pirate essaye d'installer un fichier sur votre site afin d'en prendre le contrôle (pour effacer le site, y placer des fichiers pour faire du phishing ou un script qui envoie du spam, etc.). Il cherche des trous de sécurité pour pouvoir enregistrer son fichier de prise de contrôle dans votre serveur. Si votre site web a un trou de sécurité, le pirate l'exploitera, mais comme votre site web n'a que des dossiers et fichiers interdits en écriture, il ne pourra rien enregistrer. Son attaque ne marchera pas.

Le plus simple est d'utiliser votre logiciel de FTP, d'afficher les informations relatives à un fichier ou un dossier et votre logiciel affiche l'option de modification des droits. Une autre méthode efficace, si vous avez de nombreux fichiers, est de se connecter en SSH (voir sa description plus bas). Sinon, voici un petit script PHP qui vous permettra de réaliser cette opération très simplement. Vous enregistrez ce fichier dans votre hébergement web, vous l'ouvrez depuis votre navigateur, entrez le chemin du dossier à traiter, et choisissez les réglages CHMOD pour tous les fichiers et dossiers inclus dans ce répertoire. Un rapport détaillé vous donnera les résultats. Une fois l'opération terminée, supprimez ce fichier pour éviter toute utilisation involontaire.

Vous pouvez télécharger ce fichier PHP ici: chmod.zip (2 Ko)

Code PHP:

```
<?php
/*
FORMULAIRE DE MODIFICATION DES DROITS CHMOD DES FICHIERS ET DOSSIERS
Enregistrez ce fichier dans votre répertoire hébergement web, ouvrez-le
avec votre navigateur et suivez les instructions.
Un rapport d'erreur est fourni. Supprimez le fichier après utilisation.
*/

// initialisation des variables
$dosPerm = "0";
$ficPerm = "0";
$retval = "0"; // nombre d'erreurs CHMOD

 // Chemin du dossier a traiter
    $chem = preg_replace("/[^_A-Za-z0-9-\.%\/]/i",'', $_POST["chemin"]);    // chemin de fichier a
bsolu (avec nettoyage contre piratage)
    $chem = preg_replace("/\.\./",'', $chem);    // on interdit la commande ../
    define('ABSPATH', dirname(__FILE__));
    $chem = ABSPATH.$chem;    // chemin de fichier absolu de votre compte du genre /home/log
inftp/www/ ou /home/loginftp/public_html/ etc.

//Droits des dossiers
    $d1 = preg_replace("/[^57]/",'', $_POST["dir1"]);
    $d2 = preg_replace("/[^057]/",'', $_POST["dir2"]);
    $d3 = preg_replace("/[^057]/",'', $_POST["dir3"]);
    $dosPerm = "0".$d1.$d2.$d3;
    $dosPerm = octdec($dosPerm);
//droits des fichiers
    $f1 = preg_replace("/[^46]/i",'', $_POST["fic1"]);
    $f2 = preg_replace("/[^046]/i",'', $_POST["fic2"]);
    $f3 = preg_replace("/[^046]/i",'', $_POST["fic3"]);
    $ficPerm = "0".$f1.$f2.$f3;
    $ficPerm = octdec($ficPerm);

// Formulaire html pour changer les droits
    print "<html><meta http-equiv=\"content-type\" content=\"text/html; charset=utf-8\" />";
    print "<body><h3>Changer les droits d'acc&egrave;s CHMOD aux dossiers et fichiers <br /
>dans votre h&eacute;bergement.</h3>";
    print "<table><tr><td>";
    print "<form method=\"post\">";
    print "<tr><td>Droits des dossiers: </td>";
    print "<td><select name=\"dir1\"><option value=\"5\">5</option><option value=\"7\" sel
ected>7</option></select><select name=\"dir2\"><option value=\"0\">0</option><option
```

```
value=\"5\" selected>5</option><option value=\"7\">7</option></select><select name=\"d
ir3\"><option value=\"0\">0</option><option value=\"5\" selected>5</option><option value
=\"7\">7</option></select></td></tr>";
   print "<tr><td>Droits des fichiers: </td>";
   print "<td><select name=\"fic1\"><option value=\"4\">4</option><option value=\"6\" sel
ected>6</option></select><select name=\"fic2\"><option value=\"0\">0</option><option v
alue=\"4\" selected>4</option><option value=\"6\">6</option></select><select name=\"fic
3\"><option value=\"0\">0</option><option value=\"4\" selected>4</option><option value
=\"6\">6</option></select></td></tr>";
   print "<tr><td>R&eacute;pertoire &agrave; contr&ocirc;ler: </td>";
   print "<td>".ABSPATH." <input type=\"text\" name=\"chemin\" maxlength=\"80\" size=\"30\
" value=\"/\" ></td></tr>";
   print "<tr><td> </td><td><input type=\"submit\" value=\" Changer les CHMOD des Dossi
ers et Fichiers \">";
   print "</form>";
   print "</td></tr></table>";

if ( ($dosPerm||$ficPerm) > 0 ){

   function rChmod($chem,$dosPerm,$ficPerm) {
      echo "<p><b>Journal:</b></p>\r\n";

      $d = new RecursiveDirectoryIterator($chem);
      $d ->setFlags(RecursiveDirectoryIterator::SKIP_DOTS);
      foreach (new RecursiveIteratorIterator($d, 1) as $path) {
         $chmodret = false;
         $chmodresultat = "";
         if ( $path->isDir() ) {
         $chmodret = chmod( $path, $dosPerm ); }
         else {
         if ( is_file( $path )  ) {
         $chmodret = chmod( $path, $ficPerm ); }
         }
         if ($chmodret) {$chmodresultat = "OK"; }
         else {
            $chmodresultat = "ERREUR";
            ++$retval;
            }
         echo $chmodresultat . " " . $path . "<br />\r\n";
      }
   return $retval;
   }
   $nbfailed = rChmod($chem,$dosPerm,$ficPerm);
   echo "<p><b>";
   if ($nbfailed > 0) {
      echo $nbfailed . " erreur(s) CHMOD. Voyez le journal ci-dessus.";
      }
   else echo "Pas d'erreur apparente. Vérifiez par vous-même.</b> Supprimez le fichier après u
tilisation.</p>\r\n";
}
   print "</body></html>";
?>
```

Il est possible d’accélérer le changement des droits par SSH en automatisant cette action.
Votre hébergeur doit vous donner accès à une connexion SSH.
Avec un script qui fait du pseudo-ssh en PHP, mettez le fichier dans le dossier "www" ou "public_html" et commencez le travail.

Changer tous les droits par FTP de tous les fichiers et dossiers peut être long et fastidieux avec le risque d'en oublier. J'utilise les lignes de commandes ci-dessous pour changer les droits rapidement par SSH.

Connectez-vous en SSH à votre compte, puis placez-vous dans le dossier "www" (ou "public_html") en entrant la commande cd www , et entrez les commandes suivantes en une seule ligne (après avoir modifié les noms des fichiers et dossiers selon les besoins): En mode SSH, mettez-vous dans le dossier "www" ou "public_html" avant de commencer. On copie une ligne, on appuie sur la touche Entrée, et on copie une autre ligne, on appuie sur la touche Entrée, etc. après avoir modifié les noms des fichiers et dossiers selon les besoins.

Tous les fichiers ont les droits 404 ou 444 (*droit de lecture, aucun droit d'écriture*):

```
find . -type f -print0 | xargs -0 chmod 404
```

Tous les dossiers ont les droits 505 ou 555 (*droit de lecture, aucun droit d'écriture*):

```
find . -type d -print0 | xargs -0 chmod 505
```

Tous les fichiers portant le nom ".htaccess" ont les droits 404 ou 444 (*droit de lecture, aucun droit d'écriture*):

```
find . -type f -name .htaccess -print0 | xargs -0 chmod 404
```

Tous les fichiers contenant le nom "config*.php" (utilisation du caractère joker *) du dossier "blog" ont les droits 404 (*droit de lecture, aucun droit d'écriture*):

```
find /home/loginftp/www/blog -type f -name "config*.php" -print0 | xargs -0 chmod 404
```

Tous les fichiers php ("*.php" utilisation du caractère joker *) ont les droits 404 ou 444 (*droit de lecture, aucun droit d'écriture*):

```
find . -type f -name "*.php" -print0 | xargs -0 chmod 404
```

Tous les dossiers portant le nom "dossier_a_verrouiller" ont les droits 505 ou 555 (*droit de lecture, aucun droit d'écriture*):

```
find . -type d -name dossier_a_verrouiller -print0 | xargs -0 chmod 505
```

Tous les dossiers comportant le mot upload, comme "123-upload" ou "uploadbidule" ("*upload*" utilisation du caractère joker *) qui se trouvent dans le dossier "forum" ont les droits 705 (*droit de lecture et droit d'écriture pour vous et le serveur*):

```
find /home/loginftp/www/forum -type d -name "*upload*" -print0 | xargs -0 chmod 705
```

Un article sur la signification du CHMOD et la signification des numéros.

LE FICHIER .HTACCESS

Je vous propose 10 astuces pour sécuriser votre site web. Elles sont très efficaces et permettent de stopper beaucoup de tentatives de piratages **avant** que votre CMS, blog ou e-commerce entre en action. Donc, dans une certaine mesure, si votre logiciel a une faille, peut-être que ces règles éviteront qu'elle ne soit exploitée. N'installez pas ces règles en une fois, suivez les conseils d'installations et de tests en bas de la 10è astuce.
Appliquez au moins les règles 4, 5 et 6 qui sont très efficaces, elles vous protégerons de 90% des attaques automatiques tout en ayant peu de risque de bloquer votre site internet.

Créez le fichier .htaccess avec un logiciel de texte simple (tout sauf Word). Appelez-le "txt.htaccess", envoyez-le par FTP dans votre dossier www et renommez-le en ".htaccess". Puis donnez-lui par FTP les droits 404 ou 444. Il ne sera pas modifiable. Voici une suite de commandes qui permettent de sécuriser votre site web.

1- Register_Globals sur OFF:

Si vous utilisez PHP 5.4 ou supérieur, il est inutile de s'en préoccuper. Depuis la version 4.2 de PHP, cette fonction est sur OFF et elle est obsolète depuis PHP 5.4; c'est tant mieux pour la sécurité. Or, certains hébergeurs mettent encore ce paramètre sur ON sur les serveurs mutualisés. Vérifiez cela avec votre hébergeur et corrigez-le si nécessaire.

-= INDISPENSABLE =-

Un conseil: si un script exige que le paramètre PHP Register_Globals soit sur ON, trouvez un autre script concurrent.

```
php_value register_globals OFF
// ou mettez ce code si cela ne marche pas
SetEnv REGISTER_GLOBALS 0
```

Si votre hébergeur n'utilise pas cette possibilité, alors essayez de modifier le fichier php.ini en ajoutant cette ligne:

```
register_globals = 0
```

2- Interdire l'accès à ce fichier depuis un navigateur web:

```
<Files .htaccess>
order allow,deny
deny from all
</Files>
```

3- Interdire de lister le contenu d'un dossier:

```
Options -Indexes
```

4- Exclure les logiciels suspects utilisés par les pirates et certains aspirateurs de site web.

-= TRÈS EFFICACE ET INDISPENSABLE =-

Appliquez cette règle, car on bloque déjà une grande part des attaques automatiques. Cette liste est le minimum qu'il faut avoir et elle donne d'excellents résultats. Vous pouvez en ajouter d'autres si vous en trouvez.

```
###FILTRE CONTRE CERTAINS ROBOTS DES PIRATES
RewriteEngine On
## EXCEPTION: TOUS LES ROBOTS MEMES ANONYMES OU BANNIS PEUVENT ACCEDER A CES FICHIERS
RewriteCond %{REQUEST_URI} !^/robots.txt
RewriteCond %{REQUEST_URI} !^/sitemap.xml
##
RewriteCond %{HTTP_USER_AGENT} ^-?$ [OR] ## ANONYMES
RewriteCond %{HTTP_USER_AGENT} ^curl|^Fetch\ API\ Request|GT\:\:WWW|^HTTP\:\:Lite|httplib|^Java|^LeechFTP|lwp-trivial|^LWP|libWeb|libwww|^PEAR|PECL\:\:HTTP|PHPCrawl|PycURL|python|^ReGet|Rsync|Snoopy|URI\:\:Fetch|urllib|WebDAV|^Wget [NC] ## BIBLIOTHEQUES / CLASSES HTTP DONT ON NE VEUT PAS. ATTENTION, CELA PEUT BLOQUER CERTAINES FONCTIONS DE VOTRE CMS. NE PAS TOUT EFFACER, MAIS CHERCHEZ LE NOM DE LA CLASSE HTTP CONCERNEE (DEMANDEZ AUX DEVELOPPEURS DE VOTRE CMS). CETTE LISTE BLOQUE 80% DES ROBOTS SPAMMEURS. IL FAUT LA CONSERVER.
## RewriteCond %{HTTP_USER_AGENT} ^[bcdfghjklmnpqrstvwxz\ ]{10,}|^[0-9a-z]{15,}|^[0-9A-Za-z]{19,}|^[A-Za-z]{3,}\ [a-z]{4,}\ [a-z]{4,} [OR] ## CEUX QUI INVENTENT DES NOMS AU HASARD, RETIREZ LES 2 DIESES EN DEBUT DE LIGNE POUR L'ACTIVER
RewriteRule (.*) - [F]
```

5- On bloque toute une série de failles potentielles. La plupart des pirates utilisent ces moyens pour tester la faiblesse de votre site. Là, on les bloque avant qu'il ne pénètre votre CMS, blog ou e-commerce.

-= TRÈS EFFICACE ET INDISPENSABLE =-

```
### FILTRE CONTRE XSS, REDIRECTIONS HTTP, base64_encode, VARIABLE PHP GLOBALS VIA URL, MODIFIER VARIABLE _REQUEST VIA URL, TEST DE FAILLE PHP, INJECTION SQL SIMPLE
RewriteEngine On
RewriteCond %{REQUEST_METHOD} (GET|POST) [NC]
RewriteCond %{QUERY_STRING} ^(.*)(%3C|<)/?script(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)(%3D|=)?javascript(%3A|:)(.*)$ [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)document\.location\.href(.*)$ [OR]
RewriteCond %{QUERY_STRING} ^.*(%24&x).* [NC,OR]
RewriteCond %{QUERY_STRING} ^.*(127\.0).* [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)(%3D|=)(https?|ftp|mosConfig)(%3A|:)//(.*)$ [NC,OR]
## ATTENTION A CETTE REGLE. ELLE PEUT CASSER CERTAINES REDIRECTIONS RESSEMBLANT A: http://www.truc.fr/index.php?r=http://www.google.fr ##
RewriteCond %{QUERY_STRING} ^.*(_encode|localhost|loopback).* [NC,OR]
RewriteCond %{QUERY_STRING} ^(.*)GLOBALS(=|[|%[0-9A-Z]{0,2})(.*)$ [OR]
RewriteCond %{QUERY_STRING} ^(.*)_REQUEST(=|[|%[0-9A-Z]{0,2})(.*)$ [OR]
RewriteCond %{QUERY_STRING} ^(.*)(SELECT(%20|\+)|UNION(%20|\+)ALL|INSERT(%20|\+)|DELETE(%20|\+)|CHAR\(|UPDATE(%20|\+)|REPLACE(%20|\+)|LIMIT(%20|\+)|CONCAT(%20|\+)|DECLARE(%20|\+))(.*)$ [NC]
RewriteRule (.*) - [F]
```

6- On bloque certaines requêtes bizarres:

-= TRÈS EFFICACE ET INDISPENSABLE =-

```
### DES FAUX URLS OU VIEUX SYSTEMES OBSOLETES, ON LES NEUTRALISE
RedirectMatch 403 (\.\./|base64|boot\.ini|eval\(|\(null\)|^[-_a-z0-9/\.]*//.*|/etc/passwd|^/_vti.*|
^/MSOffice.*|/fckeditor/|/elfinder/|zoho/|/jquery-file-upload/server/|/assetmanager/|wwwroot|
e107\_)
# DESACTIVE LES METHODES DE REQUETES TRACE TRACK DELETE
RewriteEngine On
RewriteCond %{REQUEST_METHOD} ^(TRACE|DELETE|TRACK) [NC]
RewriteRule ^.* - [F]
```

7- On n'autorise que l'affichage de certains fichiers, et pas les autres. Le fichier index.php est le fichier par défaut. Si on affiche index.htm, ça ne marche pas. L'intérêt est d'interdire au pirate d'afficher sur son navigateur un fichier ou un format de fichier non autorisé.

Attention: il faut tester ces interdictions et les adapter au besoin.

```
### SEUL LE FICHIER index.php EST SERVI COMME PREMIER FICHIER PAR DEFAUT. LES
AUTRES SONT INTERDITS
DirectoryIndex index.php

### INTERDIRE LES AUTRES TYPES DE FICHIER INDEX
<Files ~ "^(index)\.(p?s?x?htm?|txt|aspx?|cfml?|cgi|pl|php[3-9]|jsp|xml)$">
order allow,deny
deny from all
</Files>

### INTERDIRE L'AFFICHAGE DE CERTAINS FORMATS DE FICHIER
### EXÉCUTÉS PAR LE SERVEUR MAIS INTERDIT D'AFFICHAGE PAR LE NAVIGATEUR WEB
<Files ~ "\.(inc|class|sql|ini|conf|exe|dll|bin|tpl|bkp|dat|c|h|py|spd|theme|module|mdb|rar|
bash|git|hg|log|svn|swp|cvs)$">
deny from all
</Files>

### INTERDIRE L'AFFICHAGE DE CERTAINS FICHIERS COMME config, option, login, setup,
install, admin, home, default, xmlrpc.
### A ADAPTER SI CELA POSE PROBLEME, NOTAMMENT RETIREZ wp-(login|admin|config)| SI
VOUS UTILISEZ WORDPRESS
<Files ~ "^(wp-(login|admin|config)|config(\.inc)?|install?|admin|login|configure|
configuration|options?\.inc|option|settings?(\.inc)?|functions?(\.inc)?|setup(\.inc)?|default|
home|xmlrpc|bigdump|uploadTester|errors?|test|data|members?|hacke?r?d?|[-_a-z0-
9.]*mafia[-_a-z0-9.]*|[-_a-z0-9.]*power[-_a-z0-9.]*|[-_a-z0-9.]*jihad[-_a-z0-9.]*|php|shell|ssh|
root|cmd|[0-9]{1,6})\.(p?s?x?htm?l?|txt|aspx?|cfml?|cgi|pl|php[3-9]{0,1}|jsp?|sql|xml)$">
order allow,deny
deny from all
</Files>
```

8- Interdiction du hotlinking. Remplacez *mondomaine* par votre nom de domaine, et *\.fr* par fr, com, net, org ou autres extensions en gardant le \ avant le point.

```
### ON EVITE LE VOL D'IMAGES, VIDEO, SON, FEUILLE DE STYLE, PDF ET ZIP
### LES VISITEURS DOIVENT PASSER PAR LE SITE.
RewriteEngine on
RewriteCond %{HTTP_REFERER} !^$
RewriteCond %{HTTP_REFERER} !^https?://[-a-z0-9.]*mondomaine\.fr$ [NC]
RewriteCond %{HTTP_REFERER} !^https?://[-a-z0-9.]*mondomaine\.fr/.*$ [NC]
RewriteCond %{HTTP_REFERER} !^https?://.*(translate|paypal|google|bing|yahoo|yandex|
baidu|facebook|qwant|duck|ixquick|pinterest|twitter).*$ [NC] ## CES DOMAINES PEUVENT
AFFICHER LES ELEMENTS DU SITE
RewriteCond %{HTTP_REFERER} !^mobile?://.*$ [NC] ## LES CONNEXIONS A TRAVERS LES
APP DES MOBILES PEUVENT AFFICHER LES ELEMENTS DU SITE
RewriteRule .*\.(gif|jpe?g?|jp2|png|svgz?|css|pdf|zip|gz|js|mp3|m4a|mp4|mov|divx|avi|wma?
v?|wmp|swf|flv|docx?|xlsx?|pptx?|vbs|rtf|asf?x?|odt|ods|odp|odg|odb)$ - [NC,F]
```

9- Si des pirates ont réussi à pénétrer votre site, ils installent un script qui leur permettent de prendre les commandes de votre hébergement. Ici, on bloque la plupart des commandes de ces scripts. À tester avec votre site web, car c'est très puissant et efficace. À la 5e ligne, remplacez “/home/loginftp/” par votre chemin de fichier absolu avant le dossier “www” ou “public_html”. **Cette règle est très efficace mais peut casser votre CMS, blog ou e-commerce.** À utiliser en dernier, puis à tester intensément, et effacez éventuellement la règle qui pose problème.

```
### FILTRE CONTRE PHPSHELL.PHP, REMOTEVIEW, c99Shell et autres
RewriteEngine On
RewriteCond %{REQUEST_URI} .*((php|my)?shell|remview.*|phpremoteview.*|sshphp.*|pcom|
nstview.*|c99|r57|webadmin.*|phpget.*|phpwriter.*|fileditor.*|locus7.*|storm7.*)\.(p?s?x?htm?
l?|txt|aspx?|cfml?|cgi|pl|php[3-9]{0,1}|jsp?|sql|xml) [NC,OR]
RewriteCond %{REQUEST_METHOD} (GET|POST) [NC]
RewriteCond %{QUERY_STRING} ^(.*)=/home/loginftp/(.*)$ [OR]
RewriteCond %{QUERY_STRING} ^work_dir=.*$ [OR]
RewriteCond %{QUERY_STRING} ^command=.*&output.*$ [OR]
RewriteCond %{QUERY_STRING} ^nts_[a-z0-9_]{0,10}=.*$ [OR]
RewriteCond %{QUERY_STRING} ^(.*)cmd=.*$ [OR] ## ATTENTION A CETTE REGLE. ELLE
PEUT CASSER VOTRE SITE ##
RewriteCond %{QUERY_STRING} ^c=(t|setup|codes)$ [OR]
RewriteCond %{QUERY_STRING} ^act=((about|cmd|selfremove|chbd|trojan|backc|
massbrowsersploit|exploits|grablogins|upload.*)|((chmod|f)&f=.*))$ [OR]
RewriteCond %{QUERY_STRING} ^act=(ls|search|fsbuff|encoder|tools|processes|
ftpquickbrute|security|sql|eval|update|feedback|cmd|gofile|mkfile)&d=.*$ [OR]
RewriteCond %{QUERY_STRING} ^&?c=(l?v?i?&d=|v&fnot=|setup&ref=|l&r=|d&d=|tree&d|
t&d=|e&d=|i&d=|codes|md5crack).*$ [OR]
RewriteCond %{QUERY_STRING} ^(.*)([-_a-z]{1,15})=(ls|cd|cat|rm|mv|vim|chmod|chdir|
concat|mkdir|rmdir|pwd|clear|whoami|uname|tar|zip|unzip|gzip|gunzip|grep|more|ln|umask|
telnet|ssh|ftp|head|tail|which|mkmode|touch|logname|edit_file|search_text|find_text|
php_eval|download_file|ftp_file_down|ftp_file_up|ftp_brute|mail_file|mysql|mysql_dump|
db_query)([^a-zA-Z0-9].+)*$ [OR]
RewriteCond %{QUERY_STRING} ^(.*)(wget|shell_exec|passthru|system|exec|popen|
proc_open)(.*)$
RewriteRule (.*) - [F]
```

10- Empêcher l'exécution de tout script PHP, Perl, CGI dans un dossier. L'option ci-dessous vous permet par exemple de protéger un dossier d'upload ou tout dossier très sensible dont vous voulez renforcer la sécurité. Il ne faut pas utiliser cette option dans le fichier .htaccess avec tous les codes décrits ci-dessus. Je vous invite plutôt à créer un fichier .htaccess et à le mettre dans le dossier à protéger. Cette option empêche un navigateur web d'exécuter le script directement. Mais si le navigateur ouvre le fichier index.php qui fait un include() vers un fichier php se trouvant dans le dossier protégé par le code ci-dessous, tout s'exécutera bien. On protège donc l'exécution directe du fichier par un navigateur quand le pirate essaye d'entrer du code malicieux non filtré.

```
# Aucun script dans le dossier et ses sous-dossiers, que ce soit PHP, PERL ou autre CGI, ne pourra s'executer si ExecCGI est inactif. Et interdit d'afficher la liste des fichiers.
OPTIONS -ExecCGI  -Indexes
```

Ne mettez pas ces règles d'un coup.

Copiez-en une, puis testez votre CMS, blog ou e-commerce en ajoutant, modifiant un article, ajoutez effacez un utilisateur, accédez à votre interface d'administration et faites plusieurs choses. Si tout est OK, mettez une autre règle. En cas de problème, regardez l'URL appelée. Il y a peut-être un mot clé qui est bloqué par le fichier .htaccess. Il faudra effacer ce mot clé du fichier .htaccess. Vous l'avez compris, ce système filtre l'URL et regarde s'il est conforme à une utilisation normale. Donc, si vous avez un message d'erreur, trouvez le mot clé qui bloque la requête.
Il faut adapter ces règles à votre cas, ce n'est pas du simple copier-coller.
Plus tard, si dans l'utilisation de votre CMS, blog ou e-commerce, vous voyez une erreur 403, alors il est probable qu'une règle du filtrage soit active.
Enfin, votre CMS, blog ou e-commerce utilise souvent le fichier .htaccess pour y mettre des règles de ré-écriture d'URL plus lisibles (appelé URL rewriting). Mettez les filtres anti-pirates en premier et les règles URL rewriting à la fin. En effet, les filtres s'appliquent du premier au dernier. Placer les filtres anti-pirates après les règles de ré-écriture d'URL de votre CMS, blog ou e-commerce n'apporterait aucun bénéfice (ce n'est pas vrai à 100%, mais il y a des raisons).

AVOIR LA LISTE DES FICHIERS MODIFIÉS ET AJOUTÉS

Voici un petit script php qui vous permet d'avoir la liste des derniers fichiers créés **ET** modifiés.
Si vous avez été victime d'un piratage, il vous permettra de savoir quels fichiers ont été ajoutés et ceux qui ont été modifiés par le pirate avec la date et l'heure. Ainsi, en comparant la date de ces fichiers modifiés aux logs, vous saurez si la modification est normale ou pas et vous saurez quand et comment le pirate a frappé.
Il sert également à comprendre le comportement d'un script ou d'un CMS, blog, wiki et voir quels fichiers ont été manipulés par ce logiciel.
Copiez le code ci-dessous et créez un fichier texte que vous pourrez appeler par exemple: liste-modif.php
Mettez ce script dans votre hébergement dans le dossier "www" ou "public_html", ouvrez-le avec votre navigateur web, donnez le nombre de jours représentant la période à vérifier, puis le nom du dossier à analyser. Le chemin de fichier doit se terminer par / comme par exemple: "/forum/" qui correspondra à "/home/votreloginftp/www/forum/"
Si vous voulez vérifier tout le contenu du dossier "www" ou "public_html", cliquez uniquement sur le bouton "Vérifier Fichiers".

Attention, si vous avez beaucoup de fichiers et de répertoires, le listage peut prendre trop de temps et le script peut s'interrompre après 30s d'exécution. Si c'est le cas, essayez votre recherche répertoire par répertoire.
Ce script ne va donner la liste que des dossiers à partir du chemin "/home/votreloginftp/www/" ou "/home/votreloginftp/public_html/" de votre hébergement.
Une fois l'opération terminée, supprimez ce fichier pour éviter toute utilisation involontaire.
Vous pouvez télécharger ce fichier PHP ici: modif.zip (2,1 Ko)

Code PHP:

```php
<?php
/*
Donne la liste des derniers fichiers créés ET modifiés.
Très utile en cas de piratage pour savoir quels fichiers sont ajoutés et ceux qui ont été modifiés. Utile pour comprendre le comportement d'un script ou d'un CMS et voir quels fichiers ont été manipulés.

Mettez ce script dans votre hébergement, ouvrez-le avec votre navigateur web, donnez le nombre de jours représentant la période à vérifier, puis le nom du dossier à analyser.
Ce script ne va donner la liste que des dossiers à partir du chemin /home/votreloginftp/www/ de votre hébergement.

Supprimez le fichier après utilisation.

Crédits: Les 4/5 du code sont l'oeuvre de Linda MacPhee-Cobb (http://timestocome.com)
*/

    $go_back = 0;                    // affiche résultat ou non
    $i = 0;                          // compteur de boucle
    $dir_count = 0;                  // initialisation de la boucle
    $date = time();                  // date et heure actuelle
    $one_day = 86400;                // nombre de secondes pour une journée
    $days = preg_replace("/[^0-9]/i",'', $_POST["jours"]);    // nombre de jours à vérifier
    $path = preg_replace("/[^_A-Za-z0-9-\.%\/]/i",'', $_POST["chemin"]);    // chemin de fichier absolu (avec nettoyage contre piratage)
    $path = preg_replace("/\.\./",'', $path);    // on interdit la commande ../
    define('ABSPATH', dirname(__FILE__));
    $path = ABSPATH.$path;    // chemin de fichier absolu de votre compte du genre /home/loginftp/www/ ou /home/loginftp/public_html/ etc.
    $directories_to_read[$dir_count] = $path;

    // Formulaire pour remonter le temps
    print "<html><body><h3>Contr&ocirc;le des derniers fichiers modifi&eacute;s <br />dans votre h&eacute;bergement .</h3>";
    print "<table><tr><td>";
    print "<form method=\"post\">";
    print "<tr><td>Nombre de jours &agrave; v&eacute;rifier 1-99: </td>";
    print "<td>  <input type=\"text\" name=\"jours\" maxlength=\"2\" size=\"2\"></td></tr>";
    print "<tr><td>Nom du r&eacute;pertoire &agrave; contr&ocirc;ler: </td>";
    print "<td>".ABSPATH." <input type=\"text\" name=\"chemin\" maxlength=\"80\" size=\"30\" value=\"/\" > (mettre un / &agrave; la fin)</td></tr>";
    print "<tr><td> </td><td><input type=\"submit\" value=\" V&eacute;rifier Fichiers \">";
    print "</form>";
    print "</td></tr></table>";
    // Affichage du résultat
    $go_back = $one_day * $days;
    print "<br /> Retour sur les <strong>" . ($go_back/$one_day) ."</strong> derniers jours. <br /><br />";

    if ( $go_back > 0 ){
```

```
      print "<table><tr><th>Nom du Fichier</th><th>Date de modification</th></tr>";
      $diff = $date - $go_back;

      while ( $i <= $dir_count ){
         $current_directory = $directories_to_read[$i];

         // obtenir info fichier
         $read_path = opendir( $directories_to_read[$i] );
         while ( $file_name = readdir( $read_path)){
            if (( $file_name != '.' )&&( $file_name != '..' )){
               if ( is_dir( $current_directory . "/"  . $file_name ) == "dir" ){
                  // besoin d'obtenir tous les fichiers d'un répertoire
                  $d_file_name = "$current_directory" . "$file_name";
                  $dir_count++;
                  $directories_to_read[$dir_count] = $d_file_name . "/";
               }else{
                  $file_name = "$current_directory" . "$file_name";
                  // Si temps modifiés plus récent que x jours, affiche, sinon, passe
                  if ( (filemtime( $file_name)) > $diff  ){
                     print "<tr><td> $file_name </td>";
                     $date_changed = filemtime( $file_name );
                     $pretty_date = date("d/m/Y H:i:s", $date_changed);
                     print  "<td> ::: $pretty_date</td></tr>" ;
                  }
               }
            }
         }
         closedir ( $read_path );
         $i++;
      }
         print "</table>";
         print "<br />Supprimez le fichier après utilisation.</body></html>";
   } // if go_back > 0 )
?>
```

LES MOTS DE PASSE

Protégez vos mots de passe.
On peut essayer de pénétrer votre hébergement en devinant votre mot de passe FTP ou SQL. Si vous changez les mots de passe, respectez les règles suivantes:

1- un mot de passe doit avoir au minimum 8 caractères, plus c'est mieux.

2- il ne doit jamais être un mot qu'on trouve dans le dictionnaire d'aucune langue. Les logiciels pour cracker des mots de passe ont des dictionnaires de centaines de milliers de mots de toutes les langues et cherchent toutes les combinaisons. Cela prend entre quelques minutes à quelques petites heures pour cracker ces mots de passe très facilement.

3- Un bon mot de passe contient des lettres et des chiffres.

4- N'UTILISEZ JAMAIS LE MÊME MOT DE PASSE pour le FTP, base SQL, e-mail, interface d'administration du site web.
Le pirate **SAIT** que s'il trouve votre mot de passe, il a de fortes chances que ce soit le même mot de passe ailleurs !!!
Beaucoup d'hébergeurs proposent un mot de passe unique pour "simplifier" la gestion.
Il existe des logiciels qui proposent des mots de passe mémorisables. C'est ce qu'il y a de mieux.
Des sites web proposent aussi des mots de passe phonétiques, créant des mots faciles à mémoriser:
http://tools.arantius.com/password
http://www.freepasswordgenerator.com/
http://www.pctools.com/guides/password/

Une méthode de création de mot de passe.

Pour être sûr qu'un mot de passe mémorisable n'existe dans aucune langue, tapez-le en partie ou en entier dans Google. S'il ne retourne aucun résultat, alors votre mot de passe n'est pas un mot du dictionnaire.

INSTALLATION D'UNE BASE SQL

Lorsque vous installez votre CMS, blog ou e-commerce pour la première fois, il vous propose des réglages et paramètres par défaut que nous acceptons à chaque fois. En cas de faille, le pirate peut utiliser ces réglages et paramètres par défaut pour pénétrer votre base SQL et la modifier.
Voici quelques conseils pour éviter que cette forme d'attaque de type injection SQL soit possible. Il y a plusieurs formes d'injections SQL. La règle 5 du .htaccess en arrête une autre forme. Sinon, la vraie protection contre les injections SQL est une bonne programmation. Depuis PHP7.0, la fonction «msqli prepare» a été implantée, servez-vous-en !

1- Quand vous installez votre CMS, blog ou e-commerce, il vous donne comme login "admin" et vous demande d'entrer un mot de passe. Dans la mesure du possible, changez "admin" pour autre chose, un pseudo par exemple. Un pirate sait que le login par défaut est "admin" et lancera ses scripts uniquement sur le mot de passe. Mais si le login "admin" n'existe pas, il n'a aucune chance de pénétrer le système.
Il faut parfois faire cette modification dans phpMyadmin. Mais attention, il faut être sûr que cela ne cassera pas votre base. Posez la question sur le forum de l'éditeur de votre CMS, blog ou e-commerce pour savoir si c'est possible.

2- Le premier utilisateur est donc l'administrateur et il porte toujours le numéro (ou ID) 1. Dans le cas où le login n'est pas "admin", certains scripts peuvent essayer d'avoir le mot de passe de l'utilisateur numéro 1 qui est, dans 99,99% des cas, l'administrateur. Dans la mesure du possible, effacez l'utilisateur numéro 1 sur la liste et soyez l'administrateur portant le numéro 2.
Il faut parfois faire cette modification dans phpMyadmin. Mais attention, il faut être sûr que cela ne cassera pas votre base. Posez la question sur le forum de l'éditeur de votre CMS, blog ou e-commerce pour savoir si c'est possible.

3- Lors de l'installation, votre CMS, blog ou e-commerce vous demande de choisir un préfixe pour le nom des tables. On accepte toujours le préfixe par défaut comme wp_ pour Wordpress, g2_ pour Gallery2, dc_ pour DotClear, phpbb_ pour phpBB, etc. Le pirate peut chercher la table avec la liste des utilisateurs et leurs mots de passe. Si, comme tout le monde, vous n'avez pas changé le préfixe, il lui sera facile de trouver la table. Donc, changez le préfixe de vos tables SQL pour plus de sécurité. Vous pouvez le faire après installation. Il faut parfois faire cette modification dans phpMyadmin. Mais attention, il faut être sûr que cela ne cassera pas votre base. Posez la question sur le forum de l'éditeur de votre CMS, blog ou e-commerce pour savoir si c'est possible.
Par exemple, avec Wordpress en cas de changement de préfixe après installation, il faut aussi changer 2 entrées dans la base de données en plus du fichier wp-config.php, voyez leurs forums pour savoir comment faire.

Mon conseil:

TOUJOURS MODIFIER LES PARAMÈTRES PAR DÉFAUT !

NOMMAGE DE FICHIERS

Pour éviter que les robots des pirates ne vous trouvent grâce à Google, changez certaines habitudes et notamment le nom et l'URL de certains fichiers.

1- N'appelez pas la page de votre formulaire de contact: courrier.php ou contact.html. Appelez là autrement comme "courrier", "missive", etc. Les robots spammeurs auront plus de mal à trouver un formulaire de contact à pirater pour envoyer des spams grâce à une faille de votre script d'envoi de mail.
Faites la même chose avec d'autres fichiers: pas de login.php, admin.php, download.php (on va chercher la faille pour télécharger un fichier hors de son répertoire), etc. En règle général, évitez ces mots anglais trop répandus.

2- Les spammeurs ne sont pas idiots. Changez aussi aussi certains noms du formulaire. Dans les balises html INPUT, changez l'attribut NAME qui contient des mots comme "e-mail", "mail", "name" ou "subject" par leur équivalent en français (courriel, nom, sujet). Faites ce changement dans le formulaire HTML et dans votre script php ou cgi.

3- Évitez de donner le nom de votre CMS, blog ou e-commerce directement dans l'URL comme: www.domaine.tld/phpbb/ ou www.domaine.tld/gallery/ ou www.domaine.tld/blog/ ou www.domaine.tld/forum/ ou www.domaine.tld/admin/. Les spammeurs et piratent cherchent ces URL à cibler pour une attaque. Soyez plus original pour votre sécurité. Le mieux est d'éviter le mot anglais et de préférer son équivalent en français.

CRYPTER SON FICHIER CONFIG.INC.PHP

Malgré toutes les précautions, le pirate a pénétré votre site et cherche maintenant à connaître les login et mot de passe de votre base MySQL pour la pirater, la vider et en prendre le contrôle. On peut lui compliquer la tâche en cryptant ces données sensibles. Le serveur web pourra lire ces informations facilement, mais elles ne seront pas lisibles directement par un œil humain.
Pour un expert en PHP, cette protection ne dure que 2 minutes, cela lui fait du travail en plus, mais nous ne sommes pas là pour lui faciliter la tâche ?
Visitez ce site web et cryptez vos données.
www.phpencode.org ou
www.mobilefish.com/services/php_obfuscator/php_obfuscator.php ou cherchez un “PHP Obfuscator”.
Par exemple, mon fichier config.php contient ceci:

Code PHP:
```
<?php
//** MySQL settings **//
$db_server   = "serveursql";
$db_name     = "nombasesql";
$db_username = "loginsql";
$db_password = "motdepasse";
?>
```

Je copie la partie à encoder entre les balises <?php et ?>
Je choisis l’encodage “PHP - Extrastrength”. Ne cherchez pas un encodage plus élevé, j’ai parfois constaté des erreurs sur les serveurs web.
Je copie la longue ligne qui commence par eval(xxxx entre les balises <? et ?> et le colle dans le fichier config.inc.php, ce qui donne:

Code PHP:
```
<?php
eval(gzuncompress(gzinflate(base64_decode('AW4Akf942k3LTQqAIBQE4H3QH
QZp5cYDRDeoRXSAMHyIkFo+K7p9f5t2w3wzSkmJ7hz6Fkw5u2AZUipVFpWZRqa
0UwLQQLx5S7zOov40aE/ApyH6STP9dLsP7+LWOVoXfrZo5iMm85iP2dBTkKgv8o
MsVg==')));
?>
```

Ainsi, vous pouvez occulter toutes les informations sensibles.
Et attribuez par FTP les droits 404 ou 444 à votre fichier config.inc.php (ou équivalent) si c’est possible.

CRYPTEZ VOTRE ADRESSE E-MAIL

Si vous n'avez pas d'autre choix que d'afficher une adresse e-mail sur votre site web, vous avez 2 solutions:

1- Créez un fichier image (gif, png, jpeg) avec votre adresse écrite dessus. Ce n'est pas du texte, les robots spammeurs ne le verront pas.

2- Cryptez votre adresse e-mail avec du javascript. J'utilise cette méthode depuis des années et jamais ces adresses n'ont été spammées. Allez sur ce site web www.aspirine.org pour faire crypter votre adresse.
Pour aller encore plus loin, au lieu d'intégrer ce code à votre page html, on va l'appeler depuis un fichier javascript. L'avantage est que si l'adresse est présente sur plusieurs pages, vous n'avez à faire la modification qu'une fois.
Créez un dossier qu'on appelle "java" et on va y mettre un fichier qu'on va appeler "adresse.js". Copiez dans ce fichier la ligne de code javascript de votre adresse e-mail cryptée qui commence par "var …". Par exemple:

Code:

```
var g6="";for(var z1=0;z1<335;z1++)g6+=String.fromCharCode(("{fw%}<B\'m3xnmya\'Bwj
{tjxztrst%a\'a\'B kjwm%fA,0.a\'a\'1l4 4-jhfqujw3,?tyqnfr,aaBkjw
,ViqyVj755zaajsVnfrtistr5955zaantr,ztjxztrst%Sa\',aa,0.a\'a\'1l4V4-jhfqujw3str@5955}
(+ntrCa\',aa,aaBkjwm3xnmya\'By4-jhfqujw3,Cf4AiqyS@j7}
(+jsnfrSti.b5`ba\'a\'`1l4S\'@z5B\'\'@ktw-{fw
%u<B5@u<A}<3qjslym@u<0B88.z50B}<3xzgxyw-
u<188.3xuqny-\'\'.3wj{jwxj-.3otns-\'\'.@j{fq-z5.".charCodeAt(z1)-(-
59+64)+24+39)%(5*2+85)+-45+77);document.write(eval(g6))
```

Ensuite, dans votre page html, copiez le code suivant:

Code HTML:

```
<script src="java/adresse.js" type="text/javascript"></script>
```

ADRESSES E-MAILS À ÉVITER

Plus en rapport avec le spam qu'avec le piratage, les adresses e-mails qui ont les préfixes les plus courants sont spammées automatiquement (car ils ont plus de chances d'exister). Donc, évitez de créer des adresses avec les noms suivants:
webmaster@ admin@ contact@ email@ mail@ info@ sales@ support@ root@ www@ abuse@ news@
J'utilisais contact@ et info@ sans jamais les diffuser sur le web, mais le nombre de spams devenaient insupportables. Bref, pour le spam comme pour le piratage, il faut éviter la fainéantise intellectuelle et les paramètres par défaut.

PROTÉGER UN DOSSIER AVEC UN MOT DE PASSE

La protection par mot de passe Apache en utilisant un fichier .htaccess et un fichier .htpasswd est très efficace. Il existe plusieurs guides expliquant comment faire (par exemple ici).
Or, ils oublient tous un point important: ne nommez pas le fichier contenant le mot de passe ".htpasswd". Ce nom n'est pas obligatoire. Vous pouvez l'appeler comme vous voulez, par exemple ".htmotdepasse" ou ".htbidule". Le pirate ne trouvera pas de fichier ".htpasswd" car il est nommé autrement !

LE FICHIER ROBOTS.TXT

Vous connaissez le rôle du fichier robots.txt? http://www.robotstxt.org/ On utilise ce fichier pour interdire à un moteur de recherche d'indexer un dossier ou un fichier.
Mais ne déclarez pas dans ce fichier robots.txt un répertoire ou un fichier qui doit rester secret non seulement des moteurs de recherche mais aussi des pirates !!! Tout le monde peut lire le contenu du fichier robots.txt et donc trouver l'adresse de votre dossier/fichier secret.
Pour garder un dossier à l'abri des regards des robots indexeurs et des pirates, il est plus efficace de le protéger avec un mot de passe htaccess (voir ci-dessus).

PROTÉGER VOS FICHIERS STYLE.CSS ET INDEX.PHP

Protégez par FTP vos fichiers "style.css" et "index.php" avec les droits 404 ou 444. J'ai vu un cas de piratage du fichier style.css où le pirate avait modifié ce fichier pour afficher un pop-up.
Protégez aussi par FTP le fichier "index.php" avec les droits 404 ou 444, car c'est par lui que passe toutes les commandes de votre CMS, blog ou e-commerce.

COMMENT TESTER LA SÉCURITÉ DE MON SITE WEB ?

Voici 3 méthodes simples qui vous permettront de vérifier si vous utilisez un script avec de grosses failles de sécurité. Avec un filtrage des données entrantes et les conseils donnés dans ce forum, vous devriez être capable de régler le problème. Sinon, pour votre sécurité, changez de CMS, blog ou e-commerce car le développeur ne s'est pas préoccupé de la sécurité.
D'après mes logs, les 3 attaques suivantes représentent la grosse majorité. Ces failles sont si énormes que les pirates les cherchent en premier. Ils sont comme nous, adeptes du moindre effort.

REMARQUE 1: aucun de ces exemples ne vous apprendra à pirater un site.
Il en faut bien plus et c'est plus compliqué que ça. Cependant, vous saurez si votre script est mal codé ou non.

REMARQUE 2: Désactivez les règles de filtrages du fichier .htaccess proposées ci-dessus. En effet, ces règles vont stopper la plupart des attaques présentées ci-dessous en vous envoyant une erreur "*403 Forbidden*". Donc, pour savoir si votre script est faillible, désactivez ces règles et s'il y a une faille, corrigez le script ou installez-en un autre mieux sécurisé.

1- Attaque par exécution d'un script externe:

Créez un fichier texte contenant le code suivant:

```
<?php
echo "Hackeur vaillant, rien d'impossible !!!";
?>
```

Appelez-le "pirate.txt" et faites-en une autre copie appelée "pirate.php". Puis, par FTP, mettez-les dans le dossier www de votre serveur.
Dans votre CMS, blog ou e-commerce, les URLs ressemblent à ça (à adapter à votre cas):

```
http://www.votredomaine.tld/?page=123
```

Remplacez "123" par "http://www.votredomaine.tld/pirate.txt" comme ceci:

```
http://www.votredomaine.tld/?page=http://www.votredomaine.tld/pirate.txt?
```

Si vous voyez le message "*Hackeur vaillant, rien d'impossible !!!*" apparaître, alors vous avez un énorme trou de sécurité. Oui, un fichier texte, au lieu d'être lu, a été exécuté en php. On peut faire la même chose avec son fichier php:

```
http://www.votredomaine.tld/?page=http://www.votredomaine.tld/pirate.php?
```

On peut aller plus loin. Si le fichier pirate est sur un autre site:

```
http://www.votredomaine.tld/?page=http://www.autresiteweb.tld/pirate.txt?
```

2- Attaque par XSS ou Cross Site Scripting:

On utilise le javascript pour prendre le contrôle de votre CMS, blog ou e-commerce.
Vos URLs ressemblent à ça (à adapter à votre cas):
Code:

```
http://www.votredomaine.tld/?page=123
```

On remplace “123” par du javascript. S’il n’est pas filtré, alors c’est piraté. Par exemple:

```
http://www.votredomaine.tld/?page="><script>alert(/Hackeur vaillant, rien d’impossible !!!/)</script>
```

Ou une autre variante:

```
http://www.votredomaine.tld/?page=javascript:alert(%22Hackeur vaillant, rien d'impossible !!!%22)
```

Si une fenêtre d’alerte javascript apparaît avec le texte “*Hackeur vaillant, rien d’impossible !!!”*, votre site est ouvert à ce genre d’attaque.

3- Faille dans le téléchargement de fichiers:

Cela s’applique à 2 cas:
a) Vous avez un script de téléchargement qui propose à vos visiteurs de télécharger des fichiers. Vous avez mis tous les fichiers à télécharger dans un dossier de votre hébergement, par exemple /home/loginftp/www/download/ .
Votre URL ressemble à ça (à adapter à votre cas):

```
http://www.votredomaine.tld/download.php?fichier=monfichier.pdf
```

Si on on modifie “monfichier.pdf” qui se trouve dans le dossier “/home/loginftp/www/download/” par “../config.inc.php” qui se trouve ici “/home/loginftp/www/” comme cela:

```
http://www.votredomaine.tld/download.php?fichier=../config.inc.php
```

Est-ce qu’on le télécharge ? A-t-on ainsi les login et mot de passe de votre base de données SQL ? Croyez-le ou pas, mon script de téléchargement le permettait. Donc, on pouvait télécharger n’importe quel fichier de mon site.
b) Vous avez un script PHP qui utilise la fonction include() pour appeler d’autres fichiers. Ces fichiers sont appelés depuis l’URL et non dans le code PHP du script. Par exemple:

```
http://www.votredomaine.tld/?page=forum.php
```

Comme ci-dessus, peut-on charger d’autres fichiers? Par exemple le fichier robots.txt:

```
http://www.votredomaine.tld/?page=robots.txt
```

Normalement, on ne peut pas voir le contenu d’un fichier PHP car son code est exécuté à la différence du script de téléchargement en a). Donc, ceci devrait donner une page blanche, mais vérifiez-le quand même:

```
http://www.votredomaine.tld/?page=config.inc.php
```

Heureusement, chez certains hébergeurs la fonction PHP include() ne permet pas d'ouvrir un fichier en dehors de votre hébergement (pour des raisons de sécurité). Mais cela doit être possible sur d'autres serveurs moins sécurisés:

```
http://www.votredomaine.tld/?page=http://www.autresite.tld
```

À cœur vaillant, rien d'impossible est la devise de Jacques Coeur qui vécut au XVe s.
Comme je regrette le temps des pages statiques en html ! est ma devise !

SÉCURISER UN SCRIPT PHP

Une note très instructive sur le bon usage du PHP par la Direction centrale de la sécurité des systèmes d'information qui dépend du Secrétariat général de la défense nationale:http://www.certa.ssi.gouv.fr/site/CERTA-2007-INF-002/
Une explication sur la sécurité avec PHP est ici: http://phpsec.org/projects/guide/
Ensuite, si vous utilisez un petit script en PHP qui traite les données d'un formulaire ou d'une variable dans une URL comme "/?page=23&id=machin" (donc utilisation des requêtes GET ou POST), il faut filtrer ces données pour éviter toute faille.
J'utilise ce bout de code que je mets en tête du script afin de filtrer toutes les données qui y entrent:

```
foreach ($_REQUEST as $key => $val)
{
  $val = preg_replace("/[^_A-Za-z0-9-\.&=]/i",'', $val);
  $_REQUEST[$key] = $val;
}
```

Ainsi ne sont autorisés que les caractères alphanumériques et les signes _ - . & =
Tous les autres caractères sont effacés.
Il existe un autre filtre si celui présenté ci-dessus est trop restrictif:

```
foreach ($_REQUEST as $key => $val)
{
  $val = trim(stripslashes(htmlentities($val)));
  $_REQUEST[$key] = $val;
}
```

Pour protéger une variable précise envoyée par formulaire on peut mettre au choix un des deux filtres ci-dessous (ne pas mettre les 2 filtres pour une même variable):

```
/* pour un filtrage de base */
$variable = trim(stripslashes(htmlentities($_POST["variable"])));

/* pour un filtrage plus restrictif */
$variable = preg_replace("/[^_A-Za-z0-9-\.&=]/i",'', $_POST["variable"]);
```

Ceci est une protection générale qui fonctionne contre les formes les plus simples de piratage.
Ne mettez ce code que seulement s'il n'y a aucun système de filtrage.
À partir de PHP 5.2, il existe une série de filtres qui permet de bien cibler le filtrage des données: http://fr.php.net/filter.

SÉCURISER UN SCRIPT PERL

C'est comme pour l'article précédent sur le php. Si vous utilisez un petit script en PERL qui traite les données d'un formulaire ou d'une variable dans une URL comme "/index.cgi?page=23&id=machin" (donc utilisation des requêtes GET ou POST), il faut filtrer ces données pour éviter toute faille.
J'utilise ce bout de code que je mets en tête du script afin de filtrer toutes les données qui y entrent:

```
$val = $ENV{'QUERY_STRING'};
$val =~ s/%([a-fA-F0-9][a-fA-F0-9])/pack("C", hex($1))/eg;
$val =~ s/<([^>]|\n)*>//g;
$val =~ s/([<>\&\$;:\`\|\"\'*\?\!\~\^\(\)\[\]\{\}]|\\|\.\.|^\/)//g;
```

D'abord, on décode les caractères non alphanumériques qui sont encodés en url-encoded. Puis, on efface les caractères qui peuvent poser des problèmes comme des balises html et certains signes à la 4e ligne du code ci-dessus.
Ceci est une protection générale qui fonctionne contre les formes les plus simples de piratage.
Ne mettez ce code que seulement s'il n'y a aucun système de filtrage.

CONTRER LES INJECTIONS SQL

Cette attaque vise les serveurs web en cherchant les erreurs de programmation des scripts ASP, CGI et PHP, ces mêmes scripts exécutant des requêtes SQL. Si vous ne programmez pas, inutile de vous y attarder, il s'agit ici d'une piqure de rappel aux programmeurs.

Qu'est qu'une attaque de type SQL injection ?

SQL (Structured Query Language) est un language de base de données utilisé par nombre de CMS, blog, wiki, Gallery, etc.. C'est un language basé sur des requêtes utilisant des instructions telles que INSERT (insertion de données dans la base de données), DELETE (pour en supprimer), UPDATE (pour en mettre à jour), SELECT (pour en sélectionner et lire), et bien d'autres. Mais cette simplicité fait d'elle aussi une proie facile aux détections de failles.

Exemple de requête :

```
mysql_query("SELECT * FROM users WHERE login = 'toto';");
```

Cette requête sélectionne l'utilisateur (extrait de la table "users") dont le login est "toto". "SELECT *" signifiant qu'on sélectionne tous les champs de cette table. L'exemple ci-dessous est une attaque de type SQL injection afin de récuperer un mot de passe, c'est l'attaque la plus courante. Ma page contient un formulaire d'identification utilisateur classique où l'on entre un identifiant et un mot de passe. La requête associée à ce formulaire permettant de vérifier que les login / password entrés sont valides par rapport à notre base de données serait :
Code:

```
mysql_query("SELECT * FROM users WHERE utilisateur = '$login' AND motdepasse =
'$password'");
```

Cette requête aurait pour effet de sélectionner l'utilisateur en question si le nom d'utilisateur ET le mot de passe entrés sont dans notre base de données. Si l'un des deux est erroné, la requête ne renverra aucun résultat.

L'attaque en question
Sachant que dans mon exemple la variable $login contient ce que j'ai tapé dans mon premier champ de texte et que la variable $password contient ce que j'ai tapé dans le deuxième champ de texte (le mot de passe), voici ce qui arrivera si j'entre le code suivant dans le premier champ Identifiant (ou login) : ' OR 1=1"); // , la requête est :

```
mysql_query("SELECT * FROM users WHERE utilisateur = '' OR 1=1"); //' AND motdepasse =
'$password'");
```

Et elle permettrait certainement d'identifier l'utilisateur avec succès, car la requête est vraie si un utilisateur "" existe OU si 1=1. Comme l'utilisateur *vide* n'existe pas, mais que 1 est égal à 1, alors la requête est vraie et valide. Donc, en ajoutant ce code, on force à croire que n'importe quelle requête est vraie. Le signe "//" signifiant un commentaire en PHP, le reste du code n'est pas interprété, il est rendu inutile. Et vous voilà avec un accès à la base de données ! Ceci est l'exemple le plus simple, il en existe bien d'autres qui permettent de véritablement prendre le contrôle d'un serveur en se faisant passer pour root ou l'administrateur. Les possibilités sont malheureusement nombreuses. Nous ne les développerons pas.

Comment se protéger des attaques par injection SQL ?

Pour prévenir ces attaques, il faut connaître la programmation. Si celle-ci est bien réalisée, les attaques ne sont plus possibles. Voici quelques conseils :

- Évitez d'utiliser un compte ayant tous les pouvoirs pour l'exécution de votre serveur SQL.
- Supprimez les fonctions que vous n'utilisez pas telle que : master..xp_cmdshell, et de manière générale toutes celles commencant par "master..xp".
- Vérifiez les entrées utilisateurs telles que les champs de texte. Vérifiez aussi que les nombres attendus soient bien des nombres avec une fonction telle que IsNumeric() par exemple.
- Vérifiez aussi à bien filtrer les paramètres des URL qui sont ajoutables manuellement.
- Utilisez les caractères et fonctions d'échappement telles que AddStripSlashes() en PHP, voyez les caractéristiques de la fonction et, en général, les documentations de vos languages de programmation pour plus d'informations. Cela empêchera l'entrée utilisateur du caractère ' en l'échappant à l'aide d'un slash le précédant.
- Empêchez d'une manière générale certaines séquences d'entrées utilisateurs telles que ";", "insert", "select", "//", "--", etc.
- Limitez le nombre de caractères qu'un utilisateur peut entrer dans un champ de texte, car ceci peut fort bien lui compliquer la tâche.
- Attention à ce que vous mettez dans les cookies, car un mot de passe (même crypté en md5) est vite trouvé par une attaque de ce type. Par la suite, un remplacement de cette valeur dans le cookie évite à l'attaquant une attaque de type brute force ; c'est donc un joli cadeau.

Application concrète

La méthode la plus simple est de filtrer toutes les variables en appliquant les filtres php comme ceci :

```
/* pour un filtrage de base */
$variable = mysql_real_escape_string(htmlspecialchars($_POST["variable"]));
```

Voici un petit script pour éviter les injections SQL dans un formulaire d'identification (identifiant et mot de passe) à adapter suivant vos formulaires :

```
<?php function anti_injection( $user, $pass ) {
# on regarde s'il n'y a pas de commandes SQL
   $banlist = array (
      "insert", "select", "update", "delete", "distinct", "having", "truncate",
      "replace", "handler", "like", "procedure", "limit", "order by", "group by"
      );
   if ( eregi ( "[a-zA-Z0-9]+", $user ) ) {
      $user = trim ( str_replace ( $banlist, '', strtolower ( $user ) ) );
   } else {
      $user = NULL;
   }

   # on regarde si le mot de passe est bien alphanumérique
   # on utilise strtolower() pour faire marcher str_ireplace()
   if ( eregi ( "[a-zA-Z0-9]+", $pass ) ) {
      $pass = trim ( str_replace ( $banlist, '', strtolower ( $pass ) ) );
   } else {
      $pass = NULL;
   }

   # on fait un tableau
   # s'il y a des charactères illégaux, on arrête tout
   $array = array ( 'user' => $user, 'pass' => $pass );
   if ( in_array ( NULL, $array ) ) {
      die ( 'ERREUR : Injection SQL détectée' );
   } else {
      return $array;
   }
} // ##########
$login = anti_injection ($_POST['pseudo'],$_POST['pass']); //
on lance la fonction anti injection
$password = $login['pass']; // on recupère le pass
$password=md5($password); // on met le pass en md5
$pseudo = $login['user']; // on recupère le pseudo
?>
```

AVEZ-VOUS ÉTÉ PIRATÉ SANS LE SAVOIR ?

Vous avez un CMS, blog ou e-commerce ? Vous avez peut-être été piraté sans le savoir par des pirates qui veulent mettre des liens vers leurs sites web afin d'augmenter artificiellement leur Pagerank.
Attention, c'est très sérieux. C'est un piratage invisible, presque bénin, mais bien réel. Si vous trouvez les éléments décrits ci-dessous, votre CMS, blog ou e-commerce a été piraté par des professionnels payés pour cela, non par des gamins qui s'amusent à effacer des sites en mettant leurs pseudos à la place.
Connectez-vous à votre phpMyadmin.
Une fois identifié, dans la colonne de gauche en haut, cliquez sur le nom de votre base, puis dans la fenêtre principale, en haut cliquez sur l'onglet Recherche.
Puis entrez l'une des 3 phrases:
%display:none%
%height:0%
%visibility:hidden%
Si vous trouvez des entrées dans les textes de vos pages ou les commentaires contenant "display:none" ou "height:0" ou "visibility:hidden", alors vous avez été piraté. Ces liens web vers les sites des pirates sont invisibles mais bien présents dans le code et visibles par les robots indexeurs. Effacez ces liens, changez vos mot de passe, mettez à jour votre logiciel et suivez les conseils dans cet article.
Et si vous constatez un comportement bizarre de votre CMS, blog ou e-commerce, profitez-en pour vérifier qu'il n'y a pas de nouveaux fichiers ou dossiers ajoutés ou modifiés récemment en installant le script décrit ici-même. C'est peut-être votre CMS, blog ou e-commerce qui a fait cela, mais il vaut mieux en être sûr.