

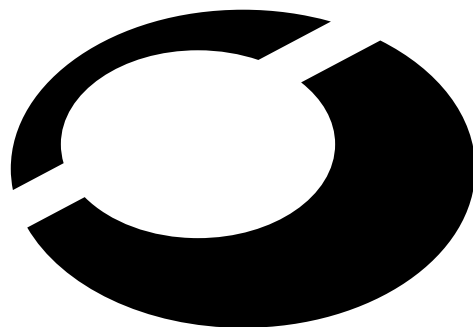
---

# PHP et la sécurité

---

Raphaël Goulais  
raphael.goulais@fr.alcove.com

version



**l'informatique est libre**

**Alcôve**

Copyright © 2000 Raphaël Goulais raphael.goulais@fr.alcove.com, Alcôve

Ce document peut être reproduit, distribué et/ou modifié selon les termes de la Licence GNU de Documentation Libre (*GNU Free Documentation Licence*) dans sa version 1.1 ou ultérieure telle que publiée, en anglais, par la *Free Software Foundation* ; sans partie invariante, avec comme première de couverture (*front cover texts*) les deux premières pages, et sans partie considérée comme quatrième de couverture (*back cover texts*)

Une copie de la licence est fournie en annexe et peut être consultée à l'url :  
[http ://www.gnu.org/copyleft/fdl.html](http://www.gnu.org/copyleft/fdl.html)

Alcôve

Centre Paris Pleyel

153 bd Anatole France

93200 Saint-Denis, France

Tél. : +33 1 49 22 68 00

Fax : +33 1 49 22 68 01

E-mail : [alcove@fr.alcove.com](mailto:alcove@fr.alcove.com), Toile : [www.alcove.com](http://www.alcove.com)

# **Table des matières**

<b>Chapitre 1 Introduction</b>	<b>3</b>
<b>Chapitre 2 Interactions avec le système</b>	<b>5</b>
<b>Chapitre 3 Surveiller votre code</b>	<b>14</b>



## Introduction



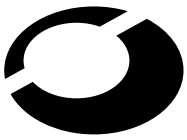
## Orientation de la présentation

### **Axes de réflexion :**

- Interactions avec le système ;
- Surveillance du code.

### **Moyens :**

- Configuration ;
- Règles d'usage.



# Interactions avec le système



## Contrôler les ressources allouées à PHP

Certaines options de configuration permettent de limiter les ressources systèmes utilisées par PHP :

- **max\_execution\_time** : temps maximal d'exécution d'un script ;
- **memory\_limit** : taille maximale de la mémoire qu'un script pourra se voir allouer ;
- **upload\_max\_file\_size** : taille maximale d'un fichier pouvant être uploadé via un formulaire.

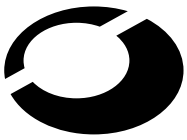


## Accès au système de fichiers

PHP offre de nombreuses options de configuration pour restreindre l'accès au système de fichiers du serveur.

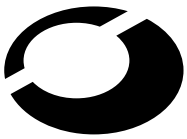
Ces options permettent de gérer les opérations autorisées sur les fichiers du système en fonction de leur propriétaire ou de leur emplacement.





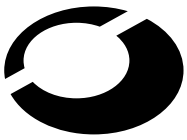
## Utilisation du Safe Mode

- **safe\_mode** : un fichier ne sera ouvert/modifié/exécuté que s'il appartient au propriétaire du script PHP exécuté ;
- **safe\_mode\_exec\_dir** : un programme ne sera pas exécuté s'il ne se trouve pas dans ce répertoire ;
- **doc\_root** : limite l'accès aux fichiers contenus dans cette arborescence (avec le safe\_mode activé) ;
- **user\_dir** : permet l'accès aux fichiers PHP d'un utilisateur, lorsque le safe\_mode est activé (typiquement, au contenu de son dossier *public\_html* ).



## Autres options

- **open\_basedir** : limite les opérations sur des fichiers au contenu de cette arborescence, indépendamment du `safe_mode` ;
- **session.save\_path** : dossier où seront stockés les fichiers de sessions ;
- **upload\_tmp\_dir** : dossier où seront stockés les fichiers uploadés ;
- **enable\_dl** : chargement dynamique d'extensions PHP ;
- **allow\_url\_fopen** : permet l'accès à des fichiers distants ;
- **disable\_function** : permet de désactiver des fonctions PHP non désirables.



## Protection de l'environnement

Certaines options limitent les possibilités de modification des variables d'environnement :

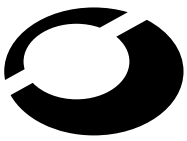
- **safe\_mode\_allowed\_env\_vars** : seule les variables commençant par ces préfixes peuvent être modifiées (en général PHP\_);
- **safe\_mode\_protected\_env\_vars** : liste de variables qu'un script ne pourra jamais modifier.



## Contrôler l'exécution de programmes externes

Toujours avec **disable\_function** , on peut désactiver les fonctions **system** , **exec** , **passthru** et **popen** qui permettent l'exécution de programmes externes.

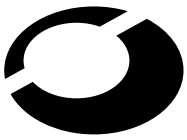
Les fonctions **escapeshellcmd** et **escapeshellarg** peuvent être utilisées pour contrôler les commandes passées au système, dans le cas où elles seraient autorisées.



## Protéger vos fichiers

Veiller à ce que le code source de vos fichiers ne puisse être consulté (en protégeant vos librairies par un `.htaccess` ou en les plaçant hors de l'arborescence du serveur, par exemple).

Une erreur courante consiste à donner l'extension `".inc"` à des fichiers destinés à être inclus par des scripts PHP. Ces fichiers, s'ils se trouvent dans l'arborescence du serveur, peuvent alors être lus par tous.



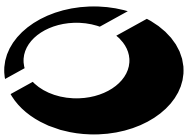
## Dissimuler PHP

La directive **expose\_php** permet d'activer ou désactiver la visibilité de PHP dans les en-têtes envoyés par le serveur.

On peut configurer le serveur pour exécuter du code PHP dans des fichiers portant une autre extension (.html .pl ou .asp par exemple).



# Surveiller votre code



## Gérer les erreurs

Les messages d'erreurs, s'ils sont visibles sur un site en production, peuvent fournir des informations précieuses sur le site hébergé. Mais ils peuvent aussi permettre pendant le développement de repérer les variables non initialisées, par exemple.

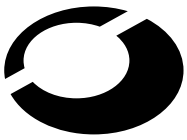
- **En cours de développement :**

```
error_reporting = E_ALL  
display_errors = On
```

- **En production :**

```
error_reporting = E_NONE  
display_errors = Off
```





### Savoir protéger ses données

L'utilisation des variables de sessions offertes par PHP permet le passage de données d'une page à l'autre sans avoir à les faire passer par des paramètres d'appel à la page.

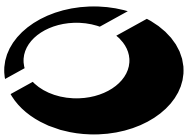
**session.referer\_check** et **session.cookie\_lifetime** permettent de contrôler la propagation de l'identification de la session. L'ajout d'un timeout sur la validité de celle-ci est un plus appréciable.



### Dangers de `register_globals`

**`register_globals`** fait passer dans l'environnement global toutes les variables passées en paramètres. L'initialisation de variable en PHP n'étant pas obligatoire, cela entraîne certains risques pour l'intégrité du code.

Malheureusement, les gains apportés par la désactivation de **`register_global`**, le sont au détriment de la compatibilité avec la plupart des scripts disponibles sur Internet.



## exemple : test.php

```
< ?php  
  
if ( check_password($login,$password) ) {  
  
$is_authenticated = 1 ;  
  
}  
  
if ( $is_authenticated == 1 ) {  
  
...  
  
}
```

[http ://myhost/test.php ?is\\_authenticated=1](http://myhost/test.php?is_authenticated=1)



## Traçabilité des variables

Les `track_vars` permettent de vérifier la provenance des données que l'on manipule.

- **`HTTP_GET_VARS`** : Variables obtenues par passage de paramètres dans l'url ;
- **`HTTP_POST_VARS`** : Variables obtenues d'un formulaire (méthode POST) ;
- **`HTTP_COOKIE_VARS`** : Variables stockées dans les cookies ;
- **`HTTP_SERVER_VARS`** : Variables du serveur ;
- **`HTTP_ENV_VARS`** : Variables d'environnement ;
- **`HTTP_SESSION_VARS`** : Variables de session ;
- **`HTTP_POST_FILES`** : Informations sur un fichier uploadé.



**Merci de votre attention**

Ce document sera disponible d'ici quelques jours sur le site internet de la société Alcôve [http ://www.alcove.com/](http://www.alcove.com/).