HERVÉ SCHAUER CONSULTANTS
Cabinet de Consultants en Sécurité Informatique depuis 1989
Spécialisé sur Unix, Windows, TCP/IP et Internet

FORUM PHP 2007

Audit de code, retour d'expérience

Nicolas Collignon
<Nicolas.Collignon@hsc.fr>

Louis Nyffenegger
<Louis.Nyffenegger@hsc.fr>

Agenda

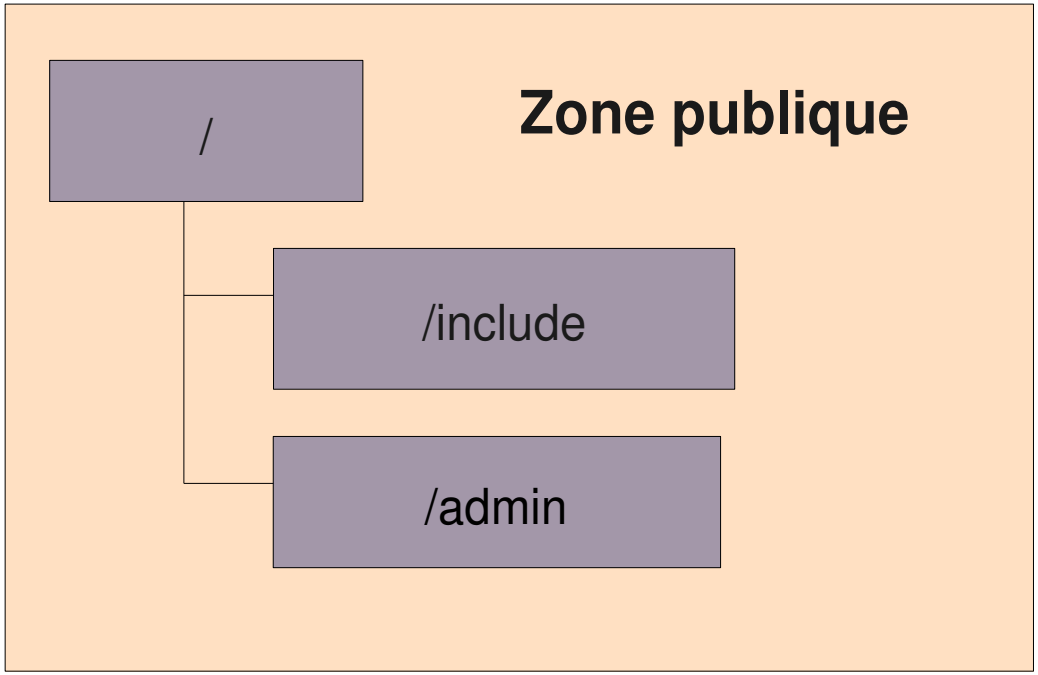
- Avant-propos
- Gestion du code
- Gestion des droits
- Gestion des erreurs
- `register_globals`
- Injection SQL
- Stockage des mots de passe
- Erreurs algorithmiques
- TOCTOU
- Utilisation du php.ini
- PHP 6
- Conclusion

Avant-propos

- Objectifs de la présentation :
  - montrer les vulnérabilités récurrentes dans les applications PHP.
  - retour d'expérience sur des audits de code PHP et des audits en boîte noire d'applications PHP.
- PHP :
  - langage facile d'accès ;
  - langage permissif.

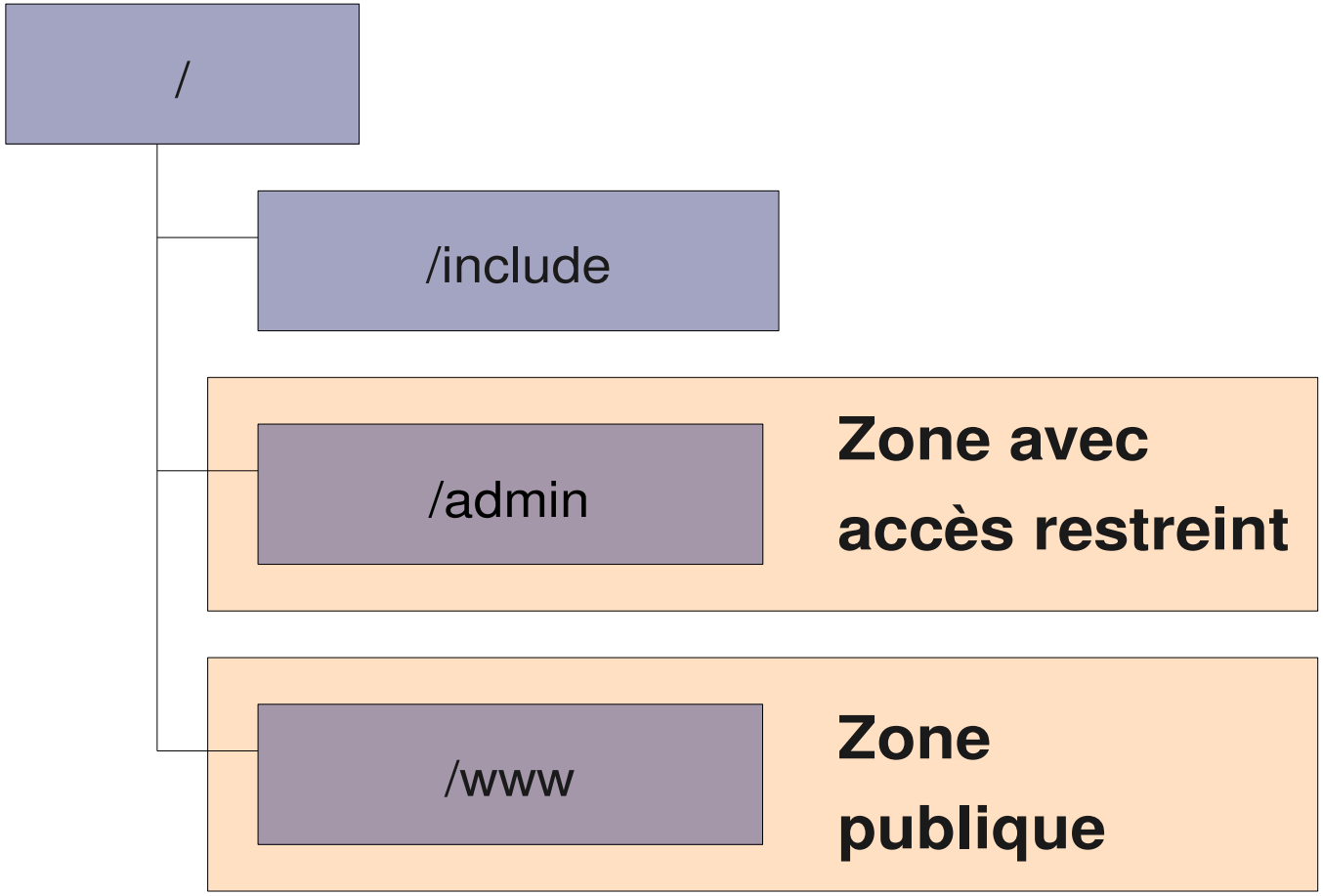
Gestion du code

- Fichiers oubliés : `phpinfo.php`, `.old`, `.sav`, `~`, `.swp`
- Fichiers non interprétés par PHP : `.inc`, `.classe`, ...
- Organisation du code non optimale :



Gestion du code

- Meilleure organisation possible :



Gestion des droits

- Pages d'administration oubliées et accessibles sans authentification

- Problèmes d'autorisation au niveau des profils
  - mon profil : http://exemples.hsc.fr/php/monprofil.php?id=100
  - un autre profil : http://exemples.hsc.fr/php/monprofil.php?id=101
  - ou gestion dans un cookie

- Souvent la gestion des droits sur les images/PDF n'est pas réalisée

Gestion des erreurs

- Une page vérifie les permissions de l'utilisateur et le redirige si l'accès est refusé.
- La modification de certains paramètres du formulaire HTML permet de générer des Warning côté serveur PHP
- Redirection impossible avec header() :
  - Possibilité de contourner l'autorisation

```
Warning: Cannot modify header information - headers
already sent by (output started at fonctions.php:1337) in
includes.php on line 11
```

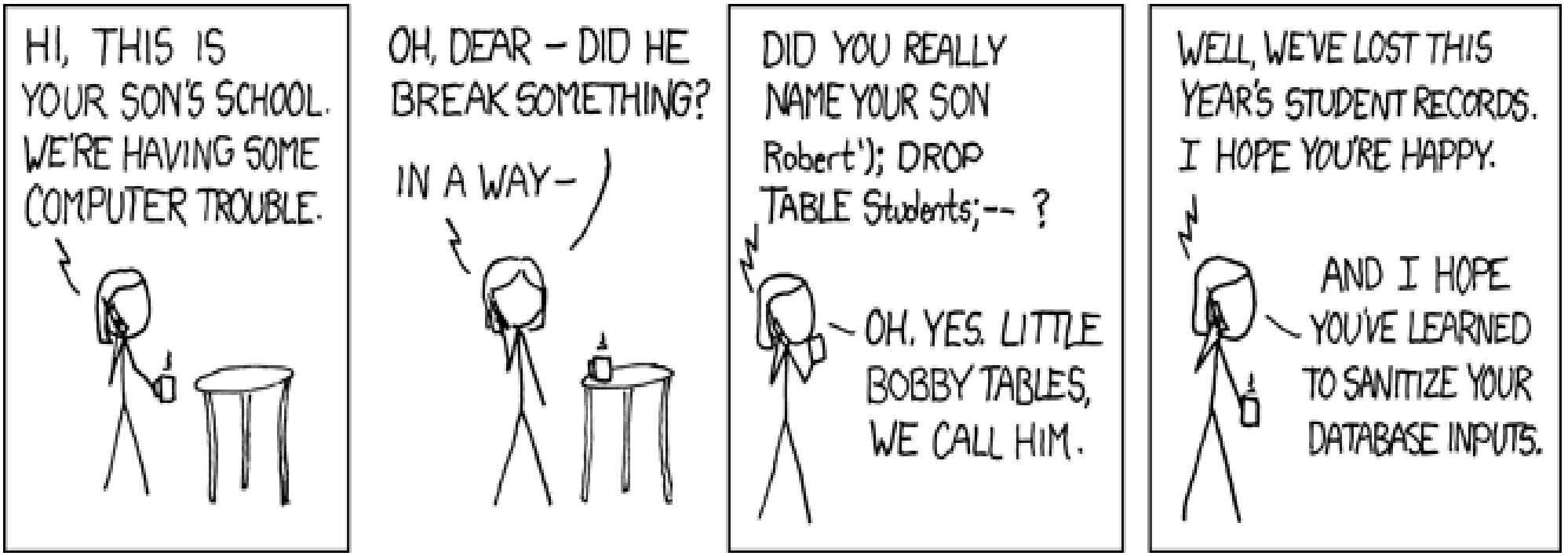
register_globals ... ou pas

- Lors d'un audit, un consultant en sécurité recommande de désactiver `register_globals`
  - Modification du fichier `php.ini`
- L'application ne fonctionne plus
- Mise en place d'une rustine

```
foreach($_REQUEST as $key => $vars)
    $GLOBALS[$key] = $vars;
```

- Rustine plus dangereuse que `register_globals`
  - Possibilité de surcharger les variables internes PHP $_XYZ ...

Injections SQL



source http://xkcd.com/327/

Injection SQL : addslashes/magic_quotes

- Souvent seul addslashes ou magic_quotes sont utilisées :
  - le fameux « `' or 1=1 /*` » ne fonctionne plus sur les formulaires d'authentification :)
  - quelques données sont pourtant oubliés :
    - variables filtrées en Javascript ;
    - champs hidden des formulaires ;
    - checkbox des formulaires.
- Cependant cette protection n'est pas suffisante :

  http://exemples.hsc.fr/php/monprofil.php?id=100 union select 1,2,3, ... /*
- Démonstration

mysqli_real_escape_string

- Bonne méthode d'encodage pour les bases MySQL
- Une mauvaise utilisation peut laisser des vulnérabilités :
  - jeu de caractères (charset) chinois / japonais
- $db->query("SET CHARACTER SET 'gbk'"); -> NOK
- $db->set_charset("gbk"); -> OK

Injection SQL : prepared statement

- Faible utilisation en comparaison à Java par exemple
- Problème du choix d'un API compatible avec plusieurs serveurs de base de données
- Attention aux mauvaises utilisations des SQL prepared statement.

Stockage des mots de passe

- Il arrive encore de trouver des mots de passe en clair
- Souvent, un simple md5 est réalisé :
  - cassable par force brute : John-The-Ripper ;
  - cassable avec des tables précalculées : Rainbowtables.
- Il est préférable d'utiliser une graine :
  - unique pour tout le site ;
  - par utilisateur.

Erreurs algorithmiques

- Commande de quantité négative :

Articles :	Prix :	Quantité :	PxQ :
Livre MySQL	**30 €**	**1**	30 €
Livre PHP	**25 €**	**-1**	**-25 €**
		TOTAL :	**5 €**

- Exemple de code vulnérable :

```
if($quantity > $availableStock) {
      $quantity = $availableStock;
}
```

  - Si `$quantity` = 1 et `$availableStock` = 3 : OK `$quantity` = 1
  - Si `$quantity` = 3 et `$availableStock` = 2 : OK `$quantity` = 2
  - Si `$quantity` = -1 et `$availableStock` = 2 : NOK `$quantity` = -1

Erreurs algorithmiques

- Correction afin d'éviter cette vulnérabilité :

```
if($quantity<0) {
        $quantity = 0;
}
if ($quantity > $availableStock) {
     $quantity = $availableStock;
}
```

  - Si `$quantity` = -1 et `$availableStock` = 2 : OK `$quantity` = 0
  - Si `$quantity` = 1 et `$availableStock` = -1 : NOK `$quantity` = **-1**

Time Of (Write|Check)/Time Of Use

- Gestion de l'image liée à un profil :

```
$name = "photo.jpg";
$fichier = fopen($name, "w");
fwrite($fichier, base64_decode($data));
fclose($fichier);

echo '<img src="'.$name.'" alt="photo" />';
```

- Que se passe-t-il lorsque 2 utilisateurs cherchent à visionner leur profil simultanément ?

Le fichier php.ini

- Utilisation de la configuration par défaut
- Quelques illusions de sécurité :
  - `magic_quotes`
  - `safe_mode`
- Possibilités de durcissement :
  - `register_globals`
  - `allow_url_include` et `allow_url_fopen`
  - `display_errors`
  - `disable_functions`

PHP6

- Modification en terme de sécurité :
  - suppression `register_globals`
  - suppression du `safe_mode`
  - suppression `magic_quotes_gpc`
- Avenir de la sécurité des applications PHP :
  - prise de conscience des développeurs ?
  - plus d'applications vulnérables sur internet ?

Questions

- **Lancement du E-learning HSC programmation sécurisée en PHP**
- **Réalisation pratique des Tests d'Intrusion Paris du 18 au 22 février 2008 :**
  - **http://www.hsc.fr/services/formations/formations_ti.html.fr**

Questions ?

Nicolas.Collignon@hsc.fr Louis.Nyffenegger@hsc.fr
www.hsc.fr