Formations en Informatique de Lille – 2016/2017

Semaine 9 - Technologies du Web

mars 2017

Sessions - Authentification - Protection de dossier

Exercice 1 : Le but de cet exercice est de construire un système de contrôle d'accès et d'authentification par login et mot de passe. Nous construirons pas à pas un script php qu'il suffira d'inclure en tête des pages PHP à protéger. La base des utilisateurs autorisés sera ici conservée dans un simple fichier de texte mais pourrait, à une petite modification près, se trouver dans une base de données.

Question 1.1 : Dans votre répertoire `public_html`, créez une sous-répertoire `tpauth` (la «racine» du présent exercice) et à l'intérieur de celui-ci un sous répertoire `lib`. Ce sous répertoire `lib` est destiné à contenir des fichiers qui n'ont pas vocation à être accessibles directement depuis le web mais à être inclus dans d'autres scripts.

Question 1.2 : Constituez dans le répertoire `lib` un fichier de texte `password.txt`. Chaque ligne de ce fichier comporte 4 parties séparées par des points virgules : login, mot de passe, nom et prénom. Par exemple

```
mallanni;animal;Malle;Annie
```

Question 1.3 : Créez dans le répertoire `lib` un fichier `Identite.class.php` comportant la définition de la classe Identité. Celle-ci contient 3 attributs destinés à recevoir des chaînes : `$login $nom $prenom` et un constructeur permettant de les initialiser.

Question 1.4 : Constituez un fichier `lib/biblio.php` qui contiendra notammment plusieurs fonctions. La première des fonctions que vous développerez s'appelle `authentifier($login, $password)`. Son résultat est soit null, soit une instance de la classe Identité. Si le login et le password fournis en argument correspondent à une ligne du fichier `password.txt`, alors le résultat est une instance de Identité représentant l'utilisateur correspondant. Sinon, le résultat vaut null.

Question 1.5 : Créez un fichier `lib/formuLogin.php` qui engendre une page HTML comportant un formulaire avec 2 champs de saisie : login, password et un bouton de validation. Les noms des variables HTTP seront `login` et `password`. Pour le traitement du formulaire, on laissera une chaîne vide (`action=""`), nous comprendrons pourquoi après (ce script n'est destiné qu'à être inclus dans un autre).

Question 1.6 : Nous utiliserons les sessions et le tableau `$_SESSION` pour distinguer si un utilisateur est identifié ou non. Pour un utilisateur identifié, `$_SESSION['ident']` contiendra un objet `Identité` le représentant. Sans authentification, `$_SESSION['ident']` sera indéfini. Lors de la phase d'authentification, si elle est réussie, on initialisera donc cette variable.
Ajouter au fichier `lib/biblio.php` une fonction `controleAuthentification()` dont le comportement est le suivant

- Si l'authentification a déjà eu lieu, la fonction termine sans rien faire
- Si des variables HTTP `login` et `password` sont disponibles dans le tableau `$_REQUEST` **ET** que ces identifiants sont corrects, alors on crée l'objet `$_SESSION['ident']` et la fonction termine normalement.
- dans **TOUS** les autres cas, la fonction déclenche une exception.

Question 1.7 : Construire un script `lib/auth.php` qui ouvre une session puis appelle la fonction précédente. Si cela ne permet pas d'authentifier l'utilisateur (la fonction a déclenché une exception), alors le script doit inclure la page formulaire `formuLogin.php` **puis déclencher l'arrêt de tout code PHP en appelant la fonction** `exit()`
Ce script sera appelé, au tout début de toute page dont on veut contrôler l'accès. En effet, si l'utilisateur est déjà authentifié, ou a envoyé un login/password correct, alors le script continuie normalement et la page protégée s'affiche. Dans le cas contraire, seul s'affiche le formulaire de connexion (l'appel à exit ayant arrêté toute exécution PHP).

Question 1.8 : Constituez dans le répertoire `tpauth` mais **hors** du répertoire `lib` une page de test qui commencera par un `require('lib/auth.php');` et affichera un message de bienvenue comportant le nom et le prénom de l'utilisateur. Vous réaliserez aussi une page `logout.php` qui effectue un `session_destroy()` afin que l'utilisateur puisse se déconnecter.

Question 1.9 : **Blocage du répertoire** `lib` : Pour sécuriser le site, c'est à dire empêcher l'accès aux fichiers sensibles rassemblés dans ce répertoire, il convient d'en interdire l'accès. Dans l'état actuel, n'importe qui peut accéder, par exemple, au fichiers des mots de passe (essayez d'entrer dans un navigateur l'URL `http://webtp.fil.univ-lille1.fr/~votrenom/tpauth/lib/password.txt`)
Vous créerez un fichier nommé (impérativement) `.htaccess` dont le contenu est

```
Order deny, allow
Deny from all
```

et le placerez dans le répertoire `lib`. Vérifiez ensuite que les fichiers du répertoire `lib` ne sont plus accessibles.