Sécurisation d'une ou plusieurs pages

Prérequis

Avoir les logiciels présentés sur Devweb 103 : Le back-end et PHP, les bases en HTML & PHP présentées sur Devweb 101 : Les bases du développement web & Devweb 103 : Le back-end et PHP.

TP 2 : Sécurisation d'une ou plusieurs pages

Un petit point sur les autorisations des fichiers

Sur tous les systèmes linux, les fichiers disposent de plusieurs propriétés pour définir qui a le droit d'en faire quoi :

- **Les propriétaires :** chaque fichier est possédé par un utilisateur et un groupe. La notation est **utilisateur:groupe**.
- Chaque fichier dispose d'autorisations propres à **l'utilisateur**, au **groupe** et aux **autres**, en :
  - **Lecture** : Le droit de lire le fichier tout simple
  - **Écriture** : Le droit de modifier les fichiers ou d'en créer de nouveaux.
  - **Exécution** : Le droit d'exécuter un fichier comme un programme.
- La notation est donc **rwxrwxrwx** pour read, write, et eXec, avec les 3 groupes différents. **rwx——** signifie que vous êtes les seuls à avoir tous les droits sur les fichiers

Un très bon résumé des droits sur les systèmes linux : http://www.tuteurs.ens.fr/unix/droits.html

Par défaut à Centrale les fichiers appartiennent à utilisateur:promo (par exemple pour un de mes fichiers : rgrondin:promo2019) et les droits sont **rwxr-xr-x**, ce qui veut dire que n'importe qui peut lire et exécuter des choses chez vous mais pas écrire.

Ceci veut dire que même en mettant un mot de passe à vos fichiers il faut en changer l'autorisation pour pas que le contenu soit dévoilé. (cc PL)

Protection par le serveur Web

Avec du .htaccess et du .htpasswd

Ce type de protection fonctionne pour protéger tout un dossier avec un nom d'utilisateur et un mot de passe.

Il faut créer deux fichiers dans le dossier :

.htaccess

```
AuthName "Secret !"
AuthType Basic
AuthUserFile "/users/promo2019/rgrondin/html/protec/.htpasswd"
Require valid-user
```

 Attention à bien changer le chemin du fichier .htpasswd

.htpasswd

```
nomdutilisateur1:motdepassecrypté1
nomdutilisateur2:motdepassecrypté2
```

 Pour crypter les mot de passes, il faut utiliser la fonction **crypt** de php

Un script simple pour les crypter :

cryptage.php

```
<?php echo crypt('motdepasse'); ?>
```

Et même pour en faire plusieurs à la suite :

cryptage.php

```
<?php
if(isset($_POST['passe'])){
   echo crypt($_POST['passe']);
}
?>
<form method="post"> <input type="password" name="passe" /> <input
type="submit" /> </form>
```

Avec CAS

 Ceci ne fonctionne que sur le perso centrale !

Encore plus simple, il suffit de rajouter ça dans un fichier .htaccess à la racine du dossier à protéger :

.htaccess

```
AuthType cas
Require valid-user
```

Tous les centraliens auront donc accès à cette page mais il existe une ruse rusée pour controller un peu les accès...

Dans vos fichiers php, vous avez accès à l'utilisateur connecté via **$_SERVER['REMOTE_USER']** donc un simple script de protection peut être :

index.php

```
<?php
if(!isset($_SESSION['REMOTE_USER']) &&
!in_array($_SESSION['REMOTE_USER'], ['rgrondin', 'pnahoum'])){
    exit('L\'accès à cette page est réservé !');
}
?>

... Un contenu hyper secret
```

Protection depuis le script

Plusieurs méthodes possible :

Protéger avec un mot de passe

Il existe plusieurs approche pour le faire mais le plus simple doit être avec une page de connexion, et des sessions pour enregistrer un utilisateur connecté.

On va utiliser une variable de session **$_SESSION['connecte']** qui va contenir le mot de passe crypté entré par l'utilisateur. On va crypter en utilisant la fonction **sha1** de PHP.

connexion.php

```
<?php
session_start();
$mdp_crypte = '5ed25af7b1ed23fb00122e13d7f74c4d8262acd8'; // Hash de
coucou

if(isset($_POST['password'])){
```

```
    $_SESSION['connecte'] = $_POST['password'];
}

if(isset($_SESSION['connecte']) && $_SESSION['connecte'] ==
$mdp_crypte){
    header('location: secret.php');
    exit;
}

if(isset($_POST['password'])) echo 'Mot de passe incorrect !';

?>

<form method="post">
Mot de passe : <input type="password" name="password" /> <br />

<input type="submit" />
</form>
```

secret.php

```
<?php
session_start();
$mdp_crypte = '5ed25af7b1ed23fb00122e13d7f74c4d8262acd8'; // Hash de
coucou

if(!isset($_SESSION['connecte']) || $_SESSION['connecte'] !=
$mdp_crypte){
    header('location: connexion.php');
    exit;
}

?>

Contenu secret...
```

Protéger avec des accès utilisateurs (login/mot de passe)

Protéger avec CAS mais depuis le script