CHIFFREMENT DE MOTS DE PASSE

1. Calcul d'une empreinte

Les mots de passe des utilisateurs ne doivent pas être stockés directement en clair dans les bases de données pour éviter les utilisations malveillantes.

C'est donc l'empreinte ou *hash* des mots de passe qui est stockée dans une base de données. Cette empreinte est calculée à l'aide d'algorithmes comme MD5, SHA-1, SHA-256, Blowfish, etc. C'est Blowfish que nous utiliserons par l'intermédiaire de la fonction *crypt* de PHP.

Vous remarquerez qu'on ne peut pas retrouver un mot de passe à partir d'une empreinte.

2. Exercice 1

Créez une base de données qui permet de gérer des utilisateurs. Les informations utiles pour chaque utilisateur sont :

- le *login* (qui sert d'identifiant) ;
- le profil (administrateur, utilisateur, comptable, etc.) ;
- l'empreinte du mot de passe obtenu avec Blowfish.

3. La fonction *crypt*

Consultez la documentation PHP concernant la fonction *crypt*.

Le sel (*salt* en anglais) permet en quelque sorte de multiplier les combinaisons de chiffrement et d'augmenter ainsi la difficulté pour un attaquant de trouver les mots de passe à partir des empreintes.

Le sel ne doit pas être stocké dans la base mais attention : la fonction *crypt* renvoie le sel et l'empreinte sous forme d'une chaîne de caractères. Il faut donc utiliser la fonction *substr* pour séparer les deux éléments.

4. Exercice 2

Écrivez le script qui permet de saisir un mot de passe et d'afficher à l'écran son empreinte.

5. Exercice 3

Écrivez le script qui permet de saisir des utilisateurs (*login*, profil, mot de passe) et de les insérer dans la base de données précédemment créée. Le script doit s'arrêter lorsque l'on tape * comme *login*.

6. Exercice 4

Écrivez le script qui permet à un utilisateur de saisir un *login* et un mot de passe.

Si l'utilisateur est authentifié (c'est-à-dire que son *login* est présent dans la base et l'empreinte du mot de passe saisi correspond à l'empreinte du mot de passe enregistrée dans la base) alors un message de bienvenue doit être affiché (par exemple, « vous êtes authentifié(e), votre profil est administrateur »).

Si la personne n'est pas authentifiée un message préviendra l'utilisateur de son échec.

7. Exercice 6 : attaque par force brute

L'attaque par force brute consiste à essayer toutes les combinaisons possibles jusqu'à trouver le bon mot de passe. Pour faire échouer l'attaque, il faut donc choisir des mots de passe suffisamment complexes et longs pour que le temps de calcul par force brute soit excessivement long même avec une grande puissance de calcul.

Bien sûr, en tant qu'administrateur système vous avez passé aux utilisateurs la consigne selon laquelle les mots de passe doivent avoir une longueur minimale de 8 caractères, et doivent être une combinaison de lettres minuscules et majuscules, de chiffres et de signes de ponctuation.

Malheureusement cette consigne n'est pas toujours respectée.

Voici un extrait des informations que l'on peut trouver dans une base de données :

```
pdurand admin uHsWBWRPdF3TrnhJuJWyyoWNz8HUkUBO
smartin utilisateur u.MP92QG1TUoMSwFavQoj6obNcTauIn2
vmarechal comptable unNGfu3UKVpjeEp9nXWLRhxSJBFt1Mci
```

Voici le sel qui a été utilisé : « $2y$10$h5qTG9D1gEpw8ZcSl2Ags7 ».

Est-ce que parmi l'extrait de la base qui vous a été fourni, des personnes ont utilisé des mots de passe simples ?

Pour le savoir, faites une recherche par force brute en n'utilisant que les lettres minuscules et une longueur de mot de passe de 3 caractères maximum.

Vous remarquerez que PHP est un langage interprété donc plutôt lent et par conséquent il n'est pas très adapté à ce genre de tâche. Il faudrait plutôt utiliser un langage compilé comme C ou C++.

8. Exercice 7 : attaque par dictionnaire

Pour accélérer les recherches, on peut éviter de tester toutes les combinaisons possibles et se concentrer sur des mots courants ou des mots de mots de passe fréquemment utilisés.

Par exemple, vous trouverez une liste de mots de passe très répandus à cette adresse : http://www.01net.com/actualites/les-plus-mauvais-mots-de-passe-2017-1335782.html.

Écrivez le script qui permet de tester si une empreinte correspond à un des mots de passe présent dans la liste fournie sur la page web citée ci-dessus.

Testez le script sur les données fournies dans l'exercice 6.