Sécurité des mots de passe

Disposer d'une gestion de mots de passe sécurisés consiste à traiter les points suivants :

Politique des mots de passe

Commencer par définir une politique de gestion des mots de passe cohérente et complète :

- Longueur : définir la longueur minimale d'un mot de passe. Aujourd'hui 12 caractères est considéré comme une valeur satisfaisante ;
- Complexité : obliger à composer son mot de passe selon différents types. Aujourd'hui, au moins un numérique, au moins une majuscule, au moins une minuscule et au moins un caractère spécial sont considérés comme une composition satisfaisante ;
- Durée de vie : obliger à changer son mot de passe régulièrement. Aujourd'hui, changer son mot de passe une à deux fois par an est considéré comme satisfaisant ;
- Seuil de blocage sur échecs répétés : limiter en nombre les tentatives d'authentification en échec. Aujourd'hui, bloquer un compte enregistrant 3 à 5 tentatives infructueuse successives est considéré comme satisfaisant ;
- Réinitialisation : définir un processus permettant ou imposant de réinitialiser son mot de passe. Fixer les conditions d'accès et les délais maximum laissés à l'utilisateur à chaque étape ;

Communication des mots de passe

L'identifiant et le mot de passe d'un utilisateur doivent lui être communiqués à la création de son compte. Considérant que ces informations sont d'une importance fondamentale qu'il convient de protéger, il sera bon de raisonner avec sérieux les moyens employés pour leur communication. Bonnes pratiques :

- Ne JAMAIS communiquer ensemble un identifiant et son mot de passe associé ;
- Si on ne peut pas éviter de communiquer les deux informations, utiliser deux canaux distincts (SMS + mail ou SMS + WhatsApp, etc.) en privilégiant un canal chiffré de bout-en-bout pour le mot de passe ;
- Lorsque l'on transmet un mot de passe, faire en sorte que celui-ci soit temporaire avec obligation de le changer à la première connexion ;
- A toute autre méthode, préférer l'envoi d'un lien temporaire d'activation qui intègrera une étape de choix du mot de passe.

Stockage des mots de passe

Hachage des mots de passe

Afin de garantir la sécurité des mots de passe, les points suivants devront être traités :

- Hacher les mots de passe au moyen d'un algorithme actualisé (actuellement hash-code à 256

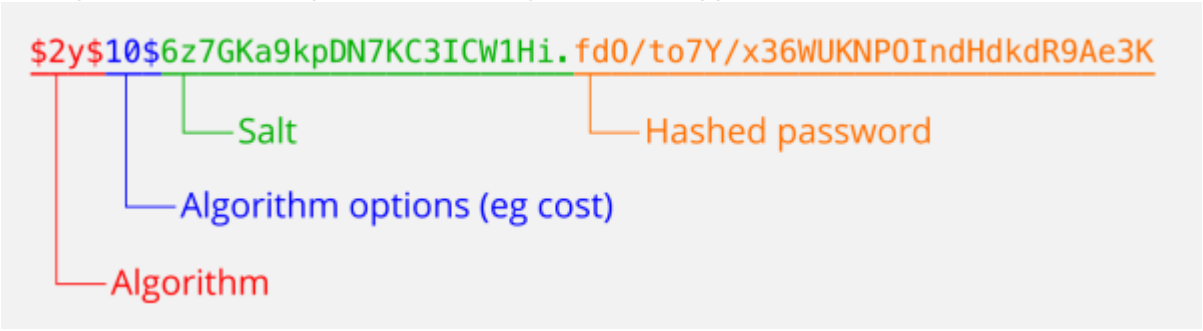
bits) ;
- Hacher les mots de passe au moyen d'un algorithme le plus lent possible (actuellement BCrypt) ;
- Hacher les mots de passe le plus tôt possible dans le traitement et supprimer définitivement et le plus tôt possible toute trace du mot de passe en clair ;
- Mémoriser les mots de passe sous leur seule forme hachée ;
- Pour l'authentification, ne JAMAIS utiliser SQL pour comparer le mot de passe saisi avec celui stocké en base de données. Le risque est d'amener le mot de passe en clair dans le SGBD où il sera enregistré dans les Logs ;
- Dans le cas général, pour l'authentification, comparer dans le langage de programmation le mot de passe saisi et fraichement haché avec le mot de passe haché stocké en base de données ;
- Une variable qui mémorise un mot de passe en clair doit avoir une durée de vie la plus courte possible. Ce qui signifie que la donnée doit être explicitement détruite ou effacée au plus vite.

Salage des mots de passe

Le hachage des mots de passe a beau être un procédé à haut niveau de sécurité lorsqu'il est convenablement mis en place (non-réversibilité, taille du hash-code élevée, algorithme lent, etc.), il n'en reste pas moins que l'ensemble comporte une fragilité particulière dans l'hypothèse du vol d'une base de données complète. Dans ce cas, un individu malveillant disposant de beaucoup de temps et d'un outillage spécialisé (**tables arc-en-ciel**) pourrait compromettre de nombreux mots de passe en s'appuyant sur l'homogénéité du hachage appliqué.

Pour contrer cette capacité, on appliquera un **salage** des mots de passe. Le salage consiste à ajouter une information aléatoire (le sel) au mot de passe avant son hachage. Le sel étant variable d'un hachage à l'autre, on aura un hash-code différent pour le hachage du même mot de passe à deux moments différents. Dans, ce cas, les tables arc-en-ciel deviennent inopérantes.

Exemple de hash-code produit avec l'algorithme BCrypt :



A titre d'exemple, PHP propose une fonction **password_hash** qui intègre un hachage BCrypt et le salage automatique.

Accompagner les utilisateurs

Les utilisateurs ne perçoivent pas nécessairement tous les enjeux liés aux mots de passe. La multiplication des comptes et les différences qui existent d'une politique de mots de passe à une autre peuvent engendrer des pratiques contre-productives. Il est donc important de réaliser un travail d'accompagnement qui s'orientera sur deux axes :

- Sensibilisation sur les dangers et les pratiques à bannir (affiches et webinaires);
- Incitation à utiliser un coffre-fort de mots de passe.