Ticket n°5
Titre du ticket : amélioration du chiffrement des mots de passe

Type du ticket : incident *(évolution/incident)*	**Niveau de gravité** : ☐Bloquant ☒ Majeur ☐ Mineur
Émetteur : Nicolas BOURGEOIS *(nom de l'émetteur)*	**Date signalement** : 27/09/2024 *(jj/mm/aaaa)*
Assignation : Loric Worms *(nom du membre de l'équipe en charge du ticket)*	**Date de résolution souhaitée** : 14/10/2024 *(jj/mm/aaaa)*
Application concernée : *R3st0.fr*	**Version** : 1.0 initiale – septembre 2024

Description du problème *(avec éventuelles captures d'écran, messages d'erreurs)* :

Par souci de renforcer la sécurité de l'application, les maîtres d'œuvre souhaitent faire évoluer le traitement des mots de passe des utilisateurs (authentification, nouvelle inscription) pour respecter les préconisations de PHP en la matière.

Actuellement, le chiffrement des mots de passe utilise la fonction crypt() avec un sel simpliste (="sel").

PHP conseille l'usage du couple de fonctions password_hash / password_verify avec l'algorithme de hachage par défaut BCRYPT.

Références :

- password_hash https://www.php.net/manual/fr/function.password-hash.php
- crypt https://www.php.net/manual/fr/function.crypt.php

Avantages de l'utilisation de password_hash :

- meilleur algorithme de hachage par défaut (BCRYPT)
- évolutive (adaptation automatique aux améliorations des algorithmes)
- salage efficace
- compatibilité avec crypt, donc les anciens mots de passe resteront utilisables, même s'il sera préférable de les modifier pour générer une meilleure empreinte.

Lexique

fonction de hachage *: calcule une empreinte numérique non réversible.*

salage *: renforce la sécurité du hachage en y ajoutant une donnée supplémentaire (le sel) afin d'empêcher que deux informations identiques conduisent à la même empreinte => protège des attaques par force brute et par table arc-en-ciel.*

coût *: rend l'algorithme arbitrairement lent et contribue à dissuader les attaques par table arc-en-ciel et par force brute.*

table arc-en-ciel *: table comportant un grand nombre d'empreintes connues, permettant de retrouver un mot de passe à partir de son empreinte.*

Solution *(diagnostic, localisation, modification, test) :*

La méthode permettant de chiffrer les mots de passe de situe dans la classe UtilisateurDAO.class.php. Ces dans cette classe que l'on doit remplacer la fonction crypt() par password_hash comme ci-dessous :

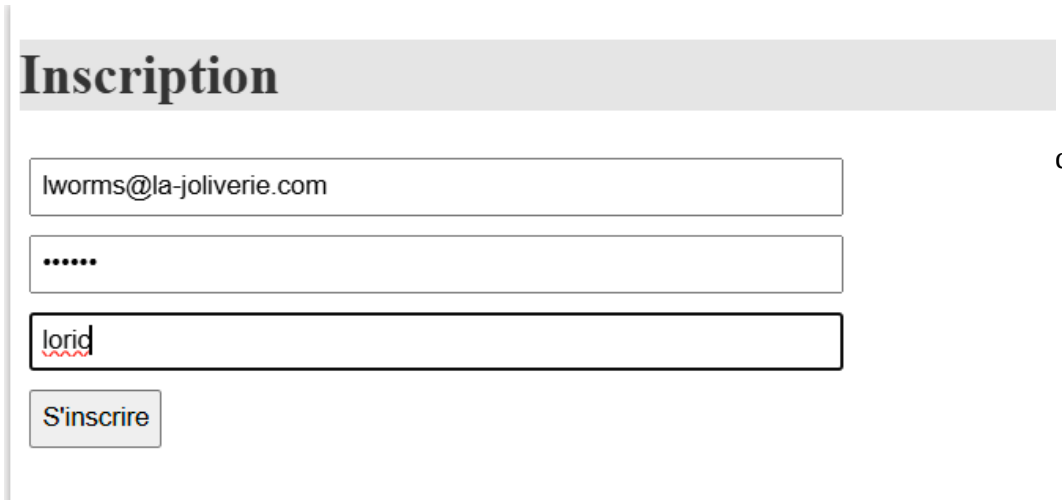
```php
public static function updateMdp(int $idU, string $mdpClair): bool {
    $ok = false;
    try {
        $requete = "UPDATE utilisateur SET mdpU = :mdpU WHERE idU = :idU";
        $stmt = Bdd::getConnexion()->prepare($requete);
        $mdpUCrypt = password_hash($mdpClair, PASSWORD_BCRYPT, ['cost' => 12]);
        $stmt->bindValue(':mdpU', $mdpUCrypt, PDO::PARAM_STR);
        $stmt->bindValue(':idU', $idU, PDO::PARAM_INT);
        $ok = $stmt->execute();
    } catch (PDOException $e) {
        throw new Exception("Erreur dans la méthode " . get_called_class() . "::updateMdp : <br/>" . $e->getMessage());
    }
    return $ok;
}
```

Par la suite il faut changer la condition qui vérifie la correspondance entre le mot de passe lors de l'authentification et le mot de passe haché dans la BDD dans la classe authentification.inc.php :

```php
        // Si le mot de passe saisi correspond au mot de passe "haché" de la BDD
        if (password_verify($mdpU, $mdpBD)) {
            // le mot de passe est celui de l'utilisateur dans la base de donnees
            $_SESSION["idU"] = $idU;        // la clef est idU désormais
            $_SESSION["mailU"] = $mailU;
            $_SESSION["mdpU"] = $mdpBD;
        }
```

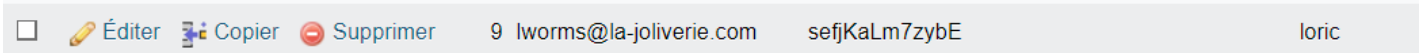
test :

Nous allons simuler une inscription comme ceci :



avec « secret » comme mot de passe.

L'utilisateur est donc créé dans la BBD avec sont mot de passe haché.



Nous allons ensuite nous connecter afin de vérifier association mot de passe claire / mot de passe haché.

La connexion est un succès, il y a bien concordance entre « secret » et « sefjKaLm7zybE ».

De plus le changement de méthode de vérification de mot de passe n’affecte pas les autres utilisateurs créer précédemment.