Ticket n°3
Titre du ticket : amélioration du chiffrement des mots de passe
Trello : https://trello.com/b/GPDB2ODV/2slamapp1g2km-ec-mb20222023

Type du ticket : incident *(évolution/incident)*	**Niveau de gravité** : ❒Bloquant ☒ Majeur ❒ Mineur
Émetteur : Nicolas BOURGEOIS *(nom de l'émetteur)*	**Date signalement** : 26/09/2022 *(jj/mm/aaaa)*
Assignation : MAINDRON Kélian *(nom du membre de l'équipe en charge du ticket)*	**Date de résolution souhaitée** : 03/10/2022 *(jj/mm/aaaa)*
Application concernée : *R3st0.fr*	**Version** : 1.0 initiale – septembre 2022

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.*

Résolution :

Pour la vérification du mot de passe en claire dans la BDD j'ai remplacé la ligne suivante

par password_verify qui est capable de détecté tous seul l'algorithme de chiffrage pour pouvoir comparer le mot de passe claire avec le mot de passe haché.

```
if (trim($mdpBD) == trim(crypt($mdpU, $mdpBD))) {
```

Pour l'inscription j'ai changé la ligne suivante par password_hash avec comme algorithme BCRYP qui est plus complexe que la fonction crypt utilisé de base.

```
$mdpUCrypt = crypt($mdpClair, "sel");
```